FPFIT, FPPLLOT and FPPAGE:
Fortran computer programs for calculating and displaying earthquake fault-plane solutions

by

P. Reasenberg and D. Oppenheimer

U.S. Geological Survey
345 Middlefield Road
Menlo Park, California

Open-File Report No. 85-739
I. FPFIT Overview

Program FPFIT finds the double-couple fault-plane solution (source model) that best fits a given set of observed first motion polarities for an earthquake. The inversion is accomplished through a two-stage grid-search procedure that finds the source model minimizing a normalized, weighted sum of first-motion polarity discrepancies. Two weighting factors are incorporated in the minimization: one reflecting the estimated variance of the data, and one based on the absolute value of the theoretical P wave radiation amplitude (Aki and Richards, 1980). The latter weighting gives greater (lesser) weight to observations near radiation lobes (nodal planes). In addition to finding the minimum-misfit solution, FPFIT finds alternative solutions corresponding to significant relative minima in misfit. Such solutions, when they exist, generally correspond to faulting mechanisms distinctly different from the minimum-misfit solution, and may be deemed the preferred solution after consideration of possible data errors, unmodeled refractions and a priori knowledge of the tectonic environment. For each double-couple source model obtained, FPFIT formally estimates the uncertainty in the model parameters (strike, dip, rake). Finally, FPFIT calculates a uniformly distributed set of solutions within the range of estimated uncertainty. This set is used in the display program FPPLOT to graphically define the range of P-axis and T-axis orientation consistent with the data.

FPFIT calculates fault-plane solutions sequentially for a suite of earthquakes and accumulates statistics on the entire suite, such as the cumulative discrepancy rates for each station and for each data class (0-weight, 1-weight, etc.). FPFIT also accumulates statistics on the suite of misfit scores, estimated solution uncertainties, and other calculated
quantities that characterize the overall quality of the inversion results. Based on an examination of these statistics, adjustments to the inversion procedure, such as revising the estimated data variance, or removing or reversing one or more individual seismograph stations, may be indicated. Such an iterative use of FPFIT provides an easy and sound way to identify and avoid some common problems in the computation of fault-plane solutions, including the presence of stations that consistently receive refracted rays that cross a nodal plane, stations with reversed polarity, and noisy data.

Certain caution must be exercised when interpreting fault-plane solutions computed by FPFIT. Non-double-couple solutions are not considered in the program, and hence adoption of a source model obtained by FPFIT implies the assumption that the earthquake is a double-couple source. Furthermore, data errors, unmodeled refractions, and oversimplified layer boundaries in the hypocentral solution should always be critically assessed in the evaluation of fault-plane solutions. Hence, we strongly recommend that all applications of FPFIT include careful consideration of reported multiple solutions, as well as a visual check for unreported alternate solutions, and that the results not be uncritically adopted.

II. Computational Procedures used in FPFIT

For each earthquake, \( E^j \), FPFIT compares the observed polarity at the \( k^{th} \) station with that calculated for a suite of source models \([M^i]\). A one-norm misfit function, \( F_{i,j} \), is defined as

\[
F_{i,j} = \frac{\sum_k \left| p_o^{j,k} - p_t^{i,k} \right| \cdot W_o^{j,k} \cdot W_t^{i,k}}{\sum_k \left| W_o^{j,k} \cdot W_t^{i,k} \right|}
\quad (1)
\]
where $p_{o,j,k}$, $p_{t,i,k}$ are terms representing, respectively, the observed and theoretical first-motion polarity (0.5 for compression, -0.5 for dilatation). The term $w_{o,j,k}$ is the observation weight that must be estimated and assigned to the data. The term

$$w_{t,i,k} = \left[ A(i,k) \right]^{1/2}$$

is the square root of the normalized theoretical P-wave radiation amplitude, $A(i,k)$, expected at the $k^{th}$ station for source model $M^i$. This weighting scheme down-weights observations near nodal planes, thereby minimizing the effect of inconsistencies near nodal planes, such as those caused by unmodeled refractions. The misfit function, $F_{i,j}^{i,j}$, is calculated for each source model in the suite, and the model that minimizes $F$ is adopted as the fault-plane solution. An exceptional case occurs when more than one solution have identical values of $F_{i,j}^{i,j}$. This case may occur because, particularly in the case of models perfectly fitting the data, the misfit function may be flat-bottomed. To break the tie in such cases FPFIT applies the additional constraint that the denominator in Equation (1) be maximized. The effect of this constraint is to maximize the distance on the focal sphere between the observations and the nodal planes.

In computation each source model is represented by the strike, dip and rake of one of its nodal planes. Specification of one plane and the associated rake angle is, of course, sufficient to represent both the fault plane and the auxiliary plane. No special importance is attached to the particular plane used to represent each model, and representation of the final solution is consequently in terms of one plane and its rake. In the companion plotting programs FPPLT and FPAGE, both planes (and "P" and "T" axes) are
calculated and displayed for each solution.

The procedure by which the suite of source models \([M^i]\) is tested is a two-stage three-dimensional grid search*. The first (coarse) stage uses 20° increments in each of the three parameters (strike, dip, and rake). All possible gridded values of rake and dip are included in the course search. However, only half the range of possible values of strike (from 0 to 160 degrees) is searched to avoid computing \(M^i\) for both the fault plane and its associated auxiliary plane. The course search identifies the solution corresponding to the minimum misfit, \(F_{\text{min}}\), and, when they exist, multiple solutions corresponding to significant relative minima in misfit. The relative minima are detected in the course search up to a level of misfit \(F \leq F_{\text{min}} + D_{\text{FITC}}\), where \(D_{\text{FITC}}\) is an input parameter. Each of these solutions is then taken as the center of a second stage (fine) 3-dimensional grid search. The fine search uses grid point spacing of 5° for strike and dip, and 10° for rake. Parameter ranges in the fine search are, relative to the central value, \(\pm 45°\) in strike and dip, \(\pm 30°\) in rake. It is in the fine grid search that the final solutions are identified, and estimates of solution parameter uncertainty are calculated. When multiple solutions are identified for an event, the solutions are distinguished by an asterisk in the output files.

* In the following discussion we assume that the default values of the search parameters are used. The default values ensure that all regions in solution space are searched. When a restricted search (that is, one in which only a subset of the possible solutions is considered) is desired, the search parameters may be explicitly set in the input control file.
III. Estimation of Parameter Uncertainty and Solution Quality

It is possible to determine formal confidence intervals for the parameters of fault-plane solutions obtained with a grid-search procedure. The misfit measure (Equation 1) is a sum of terms that take on discrete values, most of which are zero. Accordingly, the variance in $F$, and the associated 90-percent one-sided confidence interval for $F$, are estimated from the data using the method described in Appendix A. Having obtained the confidence interval for $F$, we determine the range of each solution parameter, relative to the minimum-misfit value, within which the solution misfit is bounded by the confidence interval for $F$. These ranges are taken as the uncertainties for the solution parameters.

For each fault-plane solution, FPFIT calculates the uncertainties described above and several other quantities designed to characterize the quality of the final solution. These quantities are reported, together with the fault-plane solution, in the program's output. They are:

1. $F_j = \text{minimum } [F_i,j]$, or a relative minimum of $F_i,j$. Note $F_j = 0.0$ represents a perfect fit to the data, while $F_j = 1.0$ represents a perfect misfit.

2. $\text{NOBS} = \text{number of observations used in the solution.}$

3. $\text{AVWT} = \frac{1}{\text{NOBS}} \sum_{k=1}^{\text{NOBS}} W_{o,j,k}$. AVWT is the mean data weight used in the solution; it is an overall measure of the quality of the data used in the solution. AVWT ranges from 0.0 to 30.0, with larger values reflecting
solutions computed from higher quality data.

\[
STDR = \frac{\sum_{k=1}^{\text{NOBS}} (W_{j,k} \cdot W_{i,k})}{\sum_{k=1}^{\text{NOBS}} W_{j,k}}
\]

4. STDR is the station distribution ratio. (0.0 < STDR < 1.0). This quantity is sensitive to the distribution of the data on the focal sphere, relative to the radiation pattern. When this ratio has a low value (say, STDR < 0.5), then a relatively large number of the data lie near nodal planes in the solution. Such a solution is less robust than one for which STDR > 0.5, and, consequently, should be scrutinized closely and possibly rejected.

5. ASTR, ADIP, ARAK. These quantities are ranges (in degrees) of perturbations to the strike, dip and rake, respectively, of the final solution, that result in a misfit score bounded by the 90-percent confidence interval for F. ASTR, ADIP and ARAK are taken as the uncertainties in the solution parameters. When the distribution of the data on the focal sphere do not tightly constrain one or more parameters of the solution, the corresponding uncertainties are large; when the distribution of the data tightly constrain the solution these quantities are small.

FPFIT summarizes the quality of the adopted fault-plane solution with two letter codes, each of which may be "A", "B" or "C". The first letter code, QF, summarizes the value of \( F_j \). The second quality code, QP, summarizes the three parameter uncertainties ASTR, ADIP, and ARAK, as follows:
IV. Restricted Search Mode

FPFIT may be run in either of two modes. In the "unrestricted search" mode, all possible gridded solutions are tested. This mode is the usual mode of operation for FPFIT and is the one described in Section II. However, if the user chooses to search only a subset of the possible fault plane solutions, the "restricted search" mode may be used. In this mode the user may limit the ranges of strike, dip and rake from which test solutions are drawn in the search procedure. For example, the user may specify that only thrust solutions (rake = 90 degrees), or only vertical dextral strike-slip solutions (dip = 90 degrees and rake = 180 degrees) be considered. FPFIT will search for solutions only within the specified parameter ranges.

If the restricted search mode is used, two points should be kept in mind. First, the resulting solution may not be a true minimum-misfit (zero-derivative) solution, as the adopted solution may lie at the edge of one of the restricted parameter ranges. Second, the estimates of uncertainty for the solution parameters (ASTR, ADIP, ARAK, QF and QP are invalid, owing to the
inability of FPFIT to perform its uncertainty analysis without a full range of solutions neighboring the adopted solution. Hence, the restricted search mode should only be used in special studies in which a "forced" solution is desired and estimates of solution uncertainty are not needed.

The unrestricted search mode is invoked by including in the input control file only lines 1, 2 and 3, and (optionally) station status lines, as in Appendix examples B.1.a and B.1.b (see also Section V.B).

The restricted search mode is invoked by including in the input control file an additional line specifying the ranges of the restricted search, as in Appendix examples B.1.c and B.1.d.

V. Input Files

A. Data File: FPFIT reads the print output file from program HYPO71 (Lee and Lahr, 1975). This file contains the hypocenter summary card, followed by (for each P-wave observation) the station-to-epicenter distance and azimuth, P-remark, angle of incidence, and flag denoting phase data discarded due to Jeffrey's weighting.

B. Control File (see appendix B.1): This file sets parameter values to tailor the computation to suit the particular requirements of the data set. At least three lines are required:

Line 1 - DISTMX, FMAGMN, MINOBS, IPRNT, IJEFF, NEV, DFITC (free format)

<table>
<thead>
<tr>
<th>Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTMX</td>
<td>Maximum epicentral distance in km. Phase data from stations at epicentral distances greater than DISTMX are ignored.</td>
</tr>
</tbody>
</table>
FMAGMN  Minimum magnitude. Fault-plane solutions are not computed for events with magnitudes smaller than FMAGMN.

MINOBS  Minimum number of observations. Fault plane solutions are not computed for events for which NOBS < MINOBS.

IPRNT  Print output control. If IPRNT = 1, FPFIT generates a listing of the misfit function for each test solution in the fine search. Normally, IPRNT = 0.

IJEFF  Jeffrey's weighting control. When IJEFF = 1, inversion includes phase data discarded by Jeffrey's weighting in HYPO71. When IJEFF = 0, phase data discarded by Jeffrey's weighting are omitted.

NEV  Maximum number of events for which fault plane solutions will be calculated. Solutions are calculated sequentially from the input file until NEV events have been processed.

DFITC  Depth of search for relative minima in misfit. Relative minima with $F < F_{min} + DFITC$ are considered. We recommend the value DFITC=0.05 be used for most applications.

Line 2 - ERATE(1), ERATE(2), ERATE(3), ERATE(4) (free format)

Name  Explanation

ERATE  Estimated weighted error (discrepancy) rates for classes of hand-picked data. ERATE(J+1) corresponds to assigned pick quality J. Set ERATE(J+1) = 1.0 to exclude all data from class J. For 0.0 < ERATE(J) < 0.5 lower values reflect better-quality data. Use actual weighted error rates reported in a previous run of FPFIT on same data for best estimates of ERATE.
Line 3 - ERATE(5), ERATE(6), ERATE(7), ERATE(8) (free format)

<table>
<thead>
<tr>
<th>Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERATE</td>
<td>Same as Line 2, except for classes of machine-picked (e.g., RTP) data, which are denoted by an 'X' in the first position of the P-remark field, as in XPUO. ERATE(J+5) corresponds to assigned pick quality J for machine-picked data.</td>
</tr>
</tbody>
</table>

Line 4, 5,..., N - STATUS, STATN (a1, lx, a4) (Optional)

These lines are included if data from some stations are to be omitted from the inversion, or if first motion directions from some stations are to be reversed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATUS</td>
<td>If STATUS = 'R', FPFIT reverses first motion direction for all phase data from station 'STATN'. If STATUS = 'K', FPFIT ignores all phase data from station 'STATN'.</td>
</tr>
<tr>
<td>STATN</td>
<td>Station name</td>
</tr>
</tbody>
</table>

Line N+1 - Blank (Included only if next line is present)

Line N+2 - PHI0C, PHI1, DELOC, DEL1, XLAMOC, XLAM1, DPHIC, DELC, DLAMC, DPHIF, DDELF, DLAMF (free format) (Optional)

This line is included if a restricted search range or non-default values of the grid spacing are desired; if this line is omitted, the default search range values are used, as described in section IV.
Strike is measured clockwise from north; dip is measured down from horizontal; rake of 0 = left lateral, 90 = reverse, +180 = right lateral, -90 = normal.

<table>
<thead>
<tr>
<th>Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIOC</td>
<td>Minimum value of coarse search strike range, in degrees (0 ≤ PHIOC ≤ 180)</td>
</tr>
<tr>
<td>PHI1</td>
<td>Maximum value of coarse search strike range, in degrees (0 &lt; PHI1 &lt; 180)</td>
</tr>
<tr>
<td>DELOC</td>
<td>Minimum value of coarse search dip range, in degrees (0 ≤ DELOC ≤ 90)</td>
</tr>
<tr>
<td>DEL1</td>
<td>Maximum value of coarse search dip range, in degrees (0 &lt; DEL1 &lt; 90)</td>
</tr>
<tr>
<td>XLAMOC</td>
<td>Minimum value of coarse search rake range, in degrees (-180 ≤ XLAMOC ≤ 180)</td>
</tr>
<tr>
<td>XLAM1</td>
<td>Maximum value of coarse search rake range, in degrees (-180 ≤ XLAM1 ≤ 180)</td>
</tr>
<tr>
<td>DPHIC</td>
<td>Strike increment, in degrees, for coarse search (0 &lt; DPHIC)</td>
</tr>
<tr>
<td>DDELC</td>
<td>Dip increment, in degrees, for coarse search (0 &lt; DDELC)</td>
</tr>
<tr>
<td>DLAMC</td>
<td>Rake increment, in degrees, for coarse search (0 &lt; DLAMC)</td>
</tr>
<tr>
<td>DPHIF</td>
<td>Strike increment, in degrees, for fine search (0 &lt; DPHIF)</td>
</tr>
<tr>
<td>DDELF</td>
<td>Dip increment, in degrees, for fine search (0 &lt; DDELF)</td>
</tr>
<tr>
<td>DLAMF</td>
<td>Rake increment, in degrees, for fine search (0 &lt; DLAMF)</td>
</tr>
</tbody>
</table>

VI. Output Files

A. **Extended hypocenter summary card file** (see Appendix B.2). For each earthquake satisfying the input criteria FMAGMN and MINOBS, FPFIT
calculates a fault-plane solution and writes to this file an
'extended' summary card consisting of the original HYP071 summary
card concatenated with the fault-plane solution parameters. The
format of the extended summary card is as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>HYP071 summary card.</td>
</tr>
<tr>
<td>82-84</td>
<td>Dip direction (downdip azimuth in degrees, clockwise from north).</td>
</tr>
<tr>
<td>86-87</td>
<td>Dip angle in degrees down from horizontal.</td>
</tr>
<tr>
<td>88-91</td>
<td>Rake in degrees: 0=left lateral, 90=reverse, +180=right lateral, -90=normal.</td>
</tr>
<tr>
<td>94-97</td>
<td>F_j</td>
</tr>
<tr>
<td>100-101</td>
<td>NOBS</td>
</tr>
<tr>
<td>103-107</td>
<td>AVWT</td>
</tr>
<tr>
<td>109-112</td>
<td>STDR</td>
</tr>
<tr>
<td>114-117</td>
<td>Ratio of number of machine-picked phases to total number of phases used in solution.</td>
</tr>
<tr>
<td>120-121</td>
<td>∆STR</td>
</tr>
<tr>
<td>123-124</td>
<td>∆DIP</td>
</tr>
<tr>
<td>126-127</td>
<td>∆RAK</td>
</tr>
<tr>
<td>129</td>
<td>Solution quality code QF</td>
</tr>
<tr>
<td>131</td>
<td>Solution quality code QP</td>
</tr>
<tr>
<td>132</td>
<td>Flag, normally blank. '*' indicates that solution is one of a set of multiple solutions found for this event.</td>
</tr>
</tbody>
</table>
B. **Ray file** (see Appendix B.3): This file is used as input to the plotting programs FPPLOT and FPPAGE. The first line in this file is the HYP071 heading card. Following it is, for each solution: 1) the extended hypocenter summary card; 2) the number of additional fault-plane solutions in the suite of solutions corresponding to the 90-percent confidence interval for $F_j$; 3) the dip direction, dip angle and rake of each solution in the 90-percent confidence suite (Format 11(I4,I3,I4)); 4) the phase data used in the fault-plane determination and the normalized weights assigned to them.

The format of the phase cards is as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-5</td>
<td>Station name.</td>
</tr>
<tr>
<td>7-11</td>
<td>Epicentral distance, in km.</td>
</tr>
<tr>
<td>13-17</td>
<td>Azimuth angle from epicenter to station (from north, in degrees).</td>
</tr>
<tr>
<td>19-23</td>
<td>Angle of incidence.</td>
</tr>
<tr>
<td>27-30</td>
<td>P-remark (eg., IPU2, XPDO, etc.)</td>
</tr>
<tr>
<td>32-35</td>
<td>Normalized observation weight $W_o$ used in inversion.</td>
</tr>
<tr>
<td>38</td>
<td>Discrepancy flag. If this field contains a '*', the observed polarity was discrepant with the adopted fault-plane solution. If the field is blank, observation was concordant.</td>
</tr>
</tbody>
</table>
C. **Statistical summary file** (see Appendix B.4): This file lists the options specified in the control file and used in the calculation, followed by an alphabetized summary listing of all stations used in the computation of all focal mechanisms. For each station, the summary reports the number of first motion polarity discrepancies, the number of agreements, the total number of observations, the weighted discrepancy rate (as in the numerator of Equation 1), and the fractional contribution from that station to the accumulated weighted error (misfit) from all stations for the suite of earthquakes processed. This last statistic is a measure of the effect of a station's discrepancy rate upon the entire inversion.

Following the station list are similar reports of the discrepancy rates for the hand-picked and machine-picked data, calculated separately for each data quality class (0, 1, 2, 3). The reported weighted error rate for each data class should be used as the input value for ERATE in a subsequent run of FPFIT with the same data. A double asterisk ('**') next to a reported weighted error rate indicates that this value differs from the corresponding (estimated) input value by more than 20 percent.

Next is a report of the distribution of the calculated fault-plane solution parameters $F_j$, $\Delta$DIP, $\Delta$STR, and $\Delta$RAK. These distributions provide a measure of the variance in the fault-plane set, and highlight the presence of solutions with unusually large uncertainties.

D. **Fit file** (see Appendix B.5): This file is optionally generated (when
IPRNT = 1). For each solution the extended summary card is followed by a listing of the 3-dimensional fit parameter matrix. This matrix contains the misfit scores $F_{i,j}$ (multiplied by 1000) for the source models calculated in the fine grid search, and is organized according to strike, dip and rake. The misfit score for the adopted solution is annotated with an "A"; the "additional solutions", corresponding to the 90-percent confidence interval for $F$, are annotated by an "*".

Examination of this matrix shows the behavior of $F_{i,j}$ for solutions near the adopted solution and provides some insight as to whether the adopted solution is well-constrained. This listing file consists of 3 printer pages per fault-plane solution for the default grid search range.

**VII. FPFIT Installation Considerations**

A. The dimension of all arrays are variable, and are set through assignments in the PARAMETER statement at the beginning of the main program. Thus, the array dimensions can be easily modified by changing the variable assignments in the PARAMETER statement and recompiling.

B. All I/O statements refer to variable names for logical unit numbers. These variables are also set through assignment in the PARAMETER statement. In this way, the program can be tailored to use any convenient logical unit numbers by changing the assignments in the PARAMETER statement and recompiling.

C. This program was developed on a DEC VAX 11/780 computer with a VMS operating system. The FORTRAN compiler has an extension to the FORTRAN 77 standard which permits character strings following an
exclamation point (!) anywhere on a line to be interpreted as a comment. FPFIT takes advantage of this extension to describe the function of each variable in the code on the same line as the variable declaration. These embedded comments may need to be removed for other compilers.

D. The OPEN statement for logical unit SUNIT in the main program contains the non-standard argument 'CARRIAGECONTROL=LIST'. Under VMS this changes the attributes of the file such that the printer does not interpret the first column of each line as a print control character. Similarly, all WRITE statements to logical unit SUNIT do not have carriage control characters in the first column. The OPEN statement for logical units IUNIT and CUNIT contain the non-standard argument "READONLY". Under VMS this permits the program to open files for which the user does not have "Write Status". This argument may have to be removed for use with other operating systems.

VIII. FPPLLOT Overview

FPPLLOT is an interactive plotting program for displaying fault plane solutions calculated by FPFIT. FPPLLOT produces one frame of graphic output for each solution found by FPFIT. The input file for FPPLLOT is the "RAY" output file produced by FPFIT (see section VI.B). What follows is a brief description of the graphic output from FPPLLOT. Refer to any of the examples in Appendix B.6, all of which were produced with FPPLLOT.

The top line is an image of the "Heading Card" optionally included in the HYPO71 input file used to locate the earthquake. It is reproduced here to provide a convenient label that associates the fault-plane solution with the
hypocenter location model.

Below the heading card is the extended summary card, on two lines, as described in Section VI.A.

The 5.75-inch-diameter circle below the extended summary card is a lower-hemisphere equal-area projection of the adopted fault-plane solution and first-motion data. Compressional rays are represented by circle symbols, dilational rays by triangle symbols. Upgoing (direct) rays are indicated by bold-face symbols, downgoing (refracted) rays by light-lined symbols. The size of the symbol is proportional to the observational weight \( W \) associated with the ray. The ray symbols are centered on the points of the projection to which they correspond. The bold-face symbols "P" and "T" are centered on the points corresponding to the "pressure-axis" and "tension-axis", respectively, for the adopted solution.

To the right of the 5.75-inch-diameter circle is a table listing the observations that are discrepant with the adopted fault-plane solution. The table includes, from left to right for each discrepant observation, the station name, epicentral distance, azimuth in degrees from north, angle of incidence in degrees, and the P-remark. If no observations are discrepant, this table is omitted.

The 2.5-inch-diameter circle in the lower right corner of the page is a lower-hemisphere equal-area projection showing the position of the P-axis and T-axis corresponding to the adopted fault-plane solution (bold-face "P" and "T" symbols, respectively). A set of additional pairs of P- and T-axes are plotted (light lines), corresponding to a set of solutions with misfit scores within the 90-percent confidence interval for \( F \). The resulting distribution of P- and T-axes represents the range of orientations of P and T consistent
with the data, allowing for uncertainty.

IX. FPPLOT and FPPAGE Installation Considerations

In addition to the installation considerations for FPFIT (see section VI), the following apply:

A. VMS requires that character variables be passed to plot calls by "reference". Therefore, character variables in subroutine argument lists are themselves arguments of the VMS system function \%REF. This function may be omitted for use with other operating systems.

B. The subroutine DELAY calls VMS system routines to achieve a half second delay following the clearing of the plot screen for Tektronix plotting terminals. Installation of PLOTFM on other operating systems will require the corresponding system calls.

C. All calls to graphics routines conform to standard Versatec or Calcomp software.

X. FPPAGE Overview

FPPAGE is an interactive plotting program for displaying on a single page up to 42 fault plane solutions calculated by FPFIT. The input file for FPPAGE is the "RAY" output file produced by FPFIT (see section VI.B). Refer to the example in Appendix B.7.

Each fault plane solution is represented by a lower-hemisphere equal-area projection. Above each projection is plotted a header consisting, according to the user request, of either the sequential number of the earthquake in the "RAY" file, or the origin time of the earthquake. The header is annotated
with an asterisk (*) to indicate multiple solutions. Compressional rays are depicted as solid circles; dilational rays as open circles. Plotting of the first motion symbols may be suppressed by the user. Finally the P- and T-axes of the solution are plotted. If first-motion plotting is suppressed, only the T-axis is plotted.

XI. References


ACKNOWLEDGEMENTS

Some of the computer routines used in FPFIT, FPPAGE and FPPLLOT were written by our colleagues at the U.S.G.S. in Menlo Park. John Lahr and Fred Klein kindly provided some of the graphics routines; Bruce Julian provided the moment tensor computation routines and showed us how to increase the computational speed of FPFIT with their use. Mark Mathews, at Stanford University, assisted us with the error analysis used in determining the uncertainties in the solution parameters. Donna Eberhart-Phillips assisted in the development of an early unpublished version of FPFIT.

We have had encouraging and stimulating discussions with many colleagues that were helpful in shaping these programs. We especially appreciate the comments of Jerry Eaton, Bruce Julian, John Lahr, Al Lindh, and Chris Stephens. We thank Bruce Julian and John Lahr for their reviews of the manuscript.
APPENDIX A

We wish to know the variance of $F$ in Equation (1) solely from prior knowledge of the variance of the data. To do this we take advantage of the fact that the data are provided with assigned "quality codes" (0, 1, etc.) that we assume correspond to uniform variances for each data class. This allows us to treat $F$ as a sum of binomial processes. For one earthquake, let

$n = \text{the number of observations}$

$N = \text{the number of data classes represented by the observations}$

$n_j = \text{the number of observations in data class } j$.

$$n = \sum_{j=1}^{N} n_j$$

Now let

$p_k = \text{the error (discrepancy) probability for the } k\text{'th observation}$

$r_j = \text{the error (discrepancy) rate for class } j$.

We may simplify Equation (1) by combining the weighting terms $W_0^k$ and $W_t^k$ into a single term

$$W_k = W_0^k - W_t^k$$

and expressing the misfit term

$$m_k = |p_0^k - p_t^k|$$

where

$$m_k = \begin{cases} 
1, & \text{when the } k\text{'th observation is a misfit;} \\
0, & \text{otherwise.}
\end{cases}$$
Then

\[ F = \frac{\sum_{k=1}^{n} (m_k \cdot W_k)}{\sum_{k=1}^{n} W_k} \quad (A.1) \]

The appropriate weighting terms in (A.1) are

\[ W_k = \frac{1}{\sigma_k} = \frac{1}{\sqrt{P_k(1-P_k)}} \]

Equation (A.1) can be rewritten in terms of the observations and weights in each data class as

\[ F = \frac{\sum_{j=1}^{N} \sum_{k=1}^{n_j} (m_{j,k} \cdot W_{j,k})}{\sum_{j=1}^{N} \sum_{k=1}^{n_j} W_{j,k}} \]

\[ = \frac{\sum_{j=1}^{N} \tilde{W}_j \sum_{k=1}^{n_j} m_{j,k}}{\sum_{j=1}^{N} \sum_{k=1}^{n_j} \tilde{W}_j} \]

where \( \tilde{W}_j = \frac{1}{\sigma_j} = \frac{1}{\sqrt{r_j(1-r_j)}} \).

Finally,

\[ F = \frac{\sum_{j=1}^{N} (\tilde{m}_j \cdot \tilde{W}_j)}{\sum_{j=1}^{N} (n_j \cdot \tilde{W}_j)} \quad (A.2) \]
where

\[ \tilde{m}_j = \sum_{k=1}^{n_j} m_k \]

is the number of discrepancies in data class j. Equation (A.2) is an expression for F in terms of the number of discrepancies in each data class and the (uniform) weights assigned to observations in each data class. The variance of F is easily obtained from (A.2):

\[ \text{Var}(F) = \frac{\sum_{j=1}^{N} (\text{Var}(m_j) \cdot \bar{w}_j^2)}{\left( \sum_{j=1}^{N} n_j \cdot \bar{w}_j \right)^2} \]

\[ \text{Var}(m_j) = n_j r_j (1-r_j) = n_j / \bar{w}_j^2 \]

\[ \text{Var}(F) = \frac{\sum_{j=1}^{N} n_j}{\left( \sum_{j=1}^{N} n_j \cdot \bar{w}_j \right)^2} = \frac{n}{\left( \sum_{j=1}^{N} n_j \cdot \bar{w}_j \right)^2} \]  

(A.3)

Finally, the standard deviation of F

\[ \sigma_F = \sqrt{\text{Var}(F)} \]

is calculated. The 90-percent one-sided confidence interval for F is estimated from \( \sigma_F \) by assuming that F is normally distributed.

To implement this estimation procedure we must start with some estimates of the terms \([r_j]\), recalling that these discrepancy rates include contributions from both data errors and modeling errors. While we may have some idea of the contribution to \([r_j]\) from data (reading) errors, we cannot
know the contribution from modeling (refraction) errors until after the inversion is done. Therefore the first run of FPFIT requires an educated guess for the \([r_j]\). Upon completion of the inversion for a suite of earthquakes, FPFIT reports the actual accumulated data-class error rates \([r_j]\) for the entire suite. These rates ("WEIGHTED ERROR RATE") should be used as initial estimates in the subsequent run of FPFIT. Such an iterative use of FPFIT provides a "bootstrap" ability to estimate formal error confidence limits from the data.
APPENDIX B

B.1 Examples of CONTROL FILEs for processing a set of fault-plane solutions.
B.2 Example of EXTENDED SUMMARY CARD FILE.
B.3 A portion of the RAY FILE corresponding to one selected earthquake.
B.4 STATISTICAL SUMMARY FILE resulting from a run of FPFIT on a suite of 40 earthquakes.
B.5 A portion of the FIT FILE corresponding to one selected earthquake.
B.6 Graphic output from program FPPLLOT for selected earthquakes.
B.7 Graphic output from program FP PAGE.
Examples of CONTROL FILEs for FPFIT.

a) Control file corresponding to the statistical summary file shown in B.4.
b) Control file that excludes all machine-picked data.
c) Control file that allows only vertical right-lateral slip fault planes.
d) Control file that allows only pure thrust solutions.
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Value1</th>
<th>Value2</th>
<th>Value3</th>
<th>Value4</th>
<th>Value5</th>
</tr>
</thead>
<tbody>
<tr>
<td>841422</td>
<td>154</td>
<td>5.40</td>
<td>4.50</td>
<td>3.70</td>
<td>2.90</td>
<td>2.10</td>
</tr>
<tr>
<td>841423</td>
<td>2157</td>
<td>12.27</td>
<td>11.47</td>
<td>10.67</td>
<td>9.87</td>
<td>9.07</td>
</tr>
<tr>
<td>841424</td>
<td>1859</td>
<td>16.89</td>
<td>16.09</td>
<td>15.29</td>
<td>14.49</td>
<td>13.69</td>
</tr>
<tr>
<td>841426</td>
<td>1209</td>
<td>3.90</td>
<td>3.30</td>
<td>2.70</td>
<td>2.10</td>
<td>1.50</td>
</tr>
<tr>
<td>841428</td>
<td>1733</td>
<td>3.16</td>
<td>2.66</td>
<td>2.16</td>
<td>1.66</td>
<td>1.16</td>
</tr>
<tr>
<td>841429</td>
<td>1758</td>
<td>4.36</td>
<td>3.86</td>
<td>3.36</td>
<td>2.86</td>
<td>2.36</td>
</tr>
<tr>
<td>841430</td>
<td>2568</td>
<td>4.36</td>
<td>3.86</td>
<td>3.36</td>
<td>2.86</td>
<td>2.36</td>
</tr>
<tr>
<td>841431</td>
<td>1357</td>
<td>4.36</td>
<td>3.86</td>
<td>3.36</td>
<td>2.86</td>
<td>2.36</td>
</tr>
<tr>
<td>841432</td>
<td>1922</td>
<td>4.36</td>
<td>3.86</td>
<td>3.36</td>
<td>2.86</td>
<td>2.36</td>
</tr>
<tr>
<td>841433</td>
<td>9 85</td>
<td>4.36</td>
<td>3.86</td>
<td>3.36</td>
<td>2.86</td>
<td>2.36</td>
</tr>
<tr>
<td>841434</td>
<td>839</td>
<td>3.56</td>
<td>3.16</td>
<td>2.76</td>
<td>2.36</td>
<td>1.96</td>
</tr>
<tr>
<td>841435</td>
<td>2758</td>
<td>2.76</td>
<td>2.36</td>
<td>1.96</td>
<td>1.56</td>
<td>1.16</td>
</tr>
<tr>
<td>841436</td>
<td>2259</td>
<td>2.76</td>
<td>2.36</td>
<td>1.96</td>
<td>1.56</td>
<td>1.16</td>
</tr>
<tr>
<td>841437</td>
<td>112</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841438</td>
<td>1346</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841439</td>
<td>1346</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841440</td>
<td>1934</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841441</td>
<td>1934</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841442</td>
<td>154</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841443</td>
<td>1093</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841444</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841445</td>
<td>2249</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841446</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841447</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841448</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841449</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841450</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841451</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841452</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841453</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841454</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841455</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841456</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841457</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841458</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841459</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841460</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>841461</td>
<td>1939</td>
<td>1.05</td>
<td>0.65</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>
B.3 Portion of the RAY FILE for one selected earthquake.

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>115</td>
</tr>
<tr>
<td>B</td>
<td>76</td>
</tr>
<tr>
<td>C</td>
<td>60</td>
</tr>
<tr>
<td>D</td>
<td>80</td>
</tr>
<tr>
<td>E</td>
<td>70</td>
</tr>
<tr>
<td>F</td>
<td>60</td>
</tr>
<tr>
<td>G</td>
<td>60</td>
</tr>
<tr>
<td>H</td>
<td>70</td>
</tr>
<tr>
<td>I</td>
<td>55</td>
</tr>
<tr>
<td>J</td>
<td>50</td>
</tr>
<tr>
<td>K</td>
<td>40</td>
</tr>
<tr>
<td>L</td>
<td>30</td>
</tr>
<tr>
<td>M</td>
<td>20</td>
</tr>
<tr>
<td>N</td>
<td>10</td>
</tr>
<tr>
<td>O</td>
<td>30</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
</tr>
<tr>
<td>Q</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Y</td>
<td>0</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
</tr>
</tbody>
</table>

Example values:

- Portion of the RAY FILE for one selected earthquake.
- Eighteen columns of data are shown, each containing a series of numbers and codes.
B.4 STATISTICAL SUMMARY FILE

VELEST MODEL LGARZ7 WITH REMAINING CALNET "D
CONTROL FILE = GEYS.INP;16
MAXIMUM EPICENTRAL DISTANCE = 999.MM
MINIMUM MAGNITUDE = 0.0000000E+00
MINIMUM # OBSERVATIONS = 15
PARAMETER FIT FILE SUPPRESSED (IPRNT = 0)
PHASE DATA REJECTED BY JEFFREYS' WEIGHTING EXCLUDED (IJEFF = 0)
UPTO 999 EVENTS PROCESSED
MISFIT RANGE FOR RELATIVE MINIMA IN COARSE SEARCH = 6.000000E-02

HAND-PICKED DATA: QUALITY WEIGHT WEIGHTED ERROR RATE (EST.)
# 5.1#3 8.48#
1 4.211 8.66#
2 3.333 8.10#
3 3.077 8.12#

MACHINE-PICKED DATA: QUALITY WEIGHT WEIGHTED ERROR RATE (EST.)
# 2.0#1 8.15#
1 8.### 1.###
2 8.### 1.###
3 8.### 1.###

UNRESTRICTED SEARCH RANGE:

<table>
<thead>
<tr>
<th>START STATION</th>
<th>END STATION</th>
<th>INCRNT</th>
<th>INCRNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRIKE:</td>
<td>8.8</td>
<td>168.8</td>
<td>28.8</td>
</tr>
<tr>
<td>DIP:</td>
<td>16.8</td>
<td>56.8</td>
<td>28.8</td>
</tr>
<tr>
<td>RAKE:</td>
<td>-180.0</td>
<td>168.0</td>
<td>28.8</td>
</tr>
</tbody>
</table>

**** EVENT 86#415 215# 27.52 HAS MULTIPLE SOLUTIONS ****
**** EVENT 85#907 2226 5.68 HAS MULTIPLE SOLUTIONS ****
**** EVENT 86#807 1934 16.86 HAS MULTIPLE SOLUTIONS ****
**** EVENT 85#807 1319 6.34 HAS MULTIPLE SOLUTIONS ****
**** EVENT 84#217 1334 56.85 HAS MULTIPLE SOLUTIONS ****
**** EVENT 84#616 1235 65.35 HAS MULTIPLE SOLUTIONS ****
**** EVENT 84#216 6 9.59 HAS MULTIPLE SOLUTIONS ****
**** EVENT 85#815 2047 23.99 HAS MULTIPLE SOLUTIONS ****
**** EVENT 85#714 143# 31.23 HAS MULTIPLE SOLUTIONS ****
**** EVENT 85#905 1324 19.86 HAS MULTIPLE SOLUTIONS ****
<table>
<thead>
<tr>
<th>STATION</th>
<th>DISCREPANCIES</th>
<th>AGREEMENTS</th>
<th>TOTAL WEIGHTED ERROR RATE</th>
<th>TOTAL ERROR CONTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBW</td>
<td>1</td>
<td>3</td>
<td>1.000</td>
<td>0.0001</td>
</tr>
<tr>
<td>CDU</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.000</td>
</tr>
<tr>
<td>GAR</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>0.130</td>
</tr>
<tr>
<td>GAS</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>GAR</td>
<td>5</td>
<td>197</td>
<td>282</td>
<td>0.015</td>
</tr>
<tr>
<td>GGR</td>
<td>5</td>
<td>195</td>
<td>172</td>
<td>0.010</td>
</tr>
<tr>
<td>GBM</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>GCM</td>
<td>4</td>
<td>197</td>
<td>281</td>
<td>0.085</td>
</tr>
<tr>
<td>GCR</td>
<td>5</td>
<td>288</td>
<td>285</td>
<td>0.085</td>
</tr>
<tr>
<td>GCS</td>
<td>0</td>
<td>14</td>
<td>14</td>
<td>0.000</td>
</tr>
<tr>
<td>GCV</td>
<td>6</td>
<td>167</td>
<td>173</td>
<td>0.016</td>
</tr>
<tr>
<td>GCV</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>0.000</td>
</tr>
<tr>
<td>GDC</td>
<td>3</td>
<td>53</td>
<td>56</td>
<td>0.025</td>
</tr>
<tr>
<td>GDX</td>
<td>2</td>
<td>198</td>
<td>205</td>
<td>0.013</td>
</tr>
<tr>
<td>GGL</td>
<td>0</td>
<td>36</td>
<td>37</td>
<td>0.003</td>
</tr>
<tr>
<td>GGP</td>
<td>9</td>
<td>193</td>
<td>192</td>
<td>0.015</td>
</tr>
<tr>
<td>GGU</td>
<td>8</td>
<td>21</td>
<td>21</td>
<td>0.000</td>
</tr>
<tr>
<td>GIC</td>
<td>2</td>
<td>34</td>
<td>36</td>
<td>0.022</td>
</tr>
<tr>
<td>GIC</td>
<td>6</td>
<td>37</td>
<td>43</td>
<td>0.120</td>
</tr>
<tr>
<td>GIL</td>
<td>6</td>
<td>70</td>
<td>84</td>
<td>0.055</td>
</tr>
<tr>
<td>GIL</td>
<td>8</td>
<td>11</td>
<td>11</td>
<td>0.000</td>
</tr>
<tr>
<td>GML</td>
<td>2</td>
<td>75</td>
<td>77</td>
<td>0.024</td>
</tr>
<tr>
<td>GNI</td>
<td>7</td>
<td>156</td>
<td>163</td>
<td>0.026</td>
</tr>
<tr>
<td>GNO</td>
<td>7</td>
<td>185</td>
<td>192</td>
<td>0.012</td>
</tr>
<tr>
<td>GNO</td>
<td>5</td>
<td>171</td>
<td>176</td>
<td>0.000</td>
</tr>
<tr>
<td>GNT</td>
<td>7</td>
<td>200</td>
<td>210</td>
<td>0.000</td>
</tr>
<tr>
<td>GNS</td>
<td>2</td>
<td>35</td>
<td>37</td>
<td>0.017</td>
</tr>
<tr>
<td>GSV</td>
<td>15</td>
<td>154</td>
<td>169</td>
<td>0.053</td>
</tr>
<tr>
<td>CVL</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0.000</td>
</tr>
<tr>
<td>CVM</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>0.189</td>
</tr>
<tr>
<td>JPR</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>HBP</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>0.222</td>
</tr>
<tr>
<td>HBR</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>0.338</td>
</tr>
<tr>
<td>NCF</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0.586</td>
</tr>
<tr>
<td>NDN</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>0.000</td>
</tr>
<tr>
<td>NFJ</td>
<td>1</td>
<td>10</td>
<td>19</td>
<td>0.033</td>
</tr>
<tr>
<td>NGV</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>0.282</td>
</tr>
<tr>
<td>NHB</td>
<td>18</td>
<td>93</td>
<td>109</td>
<td>0.031</td>
</tr>
<tr>
<td>NHC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>NHI</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>NLM</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>NML</td>
<td>5</td>
<td>144</td>
<td>149</td>
<td>0.028</td>
</tr>
<tr>
<td>NMT</td>
<td>9</td>
<td>124</td>
<td>133</td>
<td>0.042</td>
</tr>
<tr>
<td>NMW</td>
<td>10</td>
<td>119</td>
<td>128</td>
<td>0.056</td>
</tr>
<tr>
<td>NOL</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.359</td>
</tr>
<tr>
<td>NSG</td>
<td>11</td>
<td>158</td>
<td>161</td>
<td>0.037</td>
</tr>
<tr>
<td>NRP</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>0.004</td>
</tr>
<tr>
<td>NTH</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>0.302</td>
</tr>
<tr>
<td>NTV</td>
<td>3</td>
<td>7</td>
<td>18</td>
<td>0.316</td>
</tr>
<tr>
<td>NVA</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0.349</td>
</tr>
<tr>
<td>NVE</td>
<td>1</td>
<td>17</td>
<td>18</td>
<td>0.005</td>
</tr>
<tr>
<td>NVR</td>
<td>3</td>
<td>8</td>
<td>11</td>
<td>0.217</td>
</tr>
</tbody>
</table>

TOTAL: 189, 4111, 4300
### Summary of Hand-Picked Data with Respect to Best Fit Solutions

<table>
<thead>
<tr>
<th>Qaul</th>
<th>Discrepancies</th>
<th>Agreements</th>
<th>Total</th>
<th>Weighted Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>64</td>
<td>2576</td>
<td>2590</td>
<td>0.0106 **</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>595</td>
<td>610</td>
<td>0.0144 **</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>602</td>
<td>72</td>
<td>0.0693 **</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>178</td>
<td>191</td>
<td>0.0783 **</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>170</strong></td>
<td><strong>3973</strong></td>
<td><strong>4143</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Summary of Machine-Picked Data with Respect to Best Fit Solutions

<table>
<thead>
<tr>
<th>Qaul</th>
<th>Discrepancies</th>
<th>Agreements</th>
<th>Total</th>
<th>Weighted Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>19</td>
<td>138</td>
<td>157</td>
<td>0.0668 **</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>19</strong></td>
<td><strong>138</strong></td>
<td><strong>157</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Distribution of Solution Misfit Scores

<table>
<thead>
<tr>
<th>Misfit Score</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000 - 0.025</td>
<td>127</td>
</tr>
<tr>
<td>0.025 - 0.050</td>
<td>41</td>
</tr>
<tr>
<td>0.050 - 0.075</td>
<td>27</td>
</tr>
<tr>
<td>0.075 - 0.100</td>
<td>13</td>
</tr>
<tr>
<td>0.100 - 0.125</td>
<td>2</td>
</tr>
<tr>
<td>0.125 - 0.150</td>
<td>1</td>
</tr>
<tr>
<td>0.150 - 0.175</td>
<td>1</td>
</tr>
<tr>
<td>0.175 - 0.200</td>
<td>1</td>
</tr>
<tr>
<td>0.200 - 0.225</td>
<td>1</td>
</tr>
<tr>
<td>0.225 - 0.250</td>
<td>1</td>
</tr>
<tr>
<td>0.250 - 0.275</td>
<td>1</td>
</tr>
<tr>
<td>0.275 - 0.300</td>
<td>6</td>
</tr>
<tr>
<td>0.300 - 0.325</td>
<td>6</td>
</tr>
<tr>
<td>0.325 - 0.350</td>
<td>6</td>
</tr>
<tr>
<td>0.350 - 0.375</td>
<td>6</td>
</tr>
<tr>
<td>0.375 - 0.400</td>
<td>6</td>
</tr>
<tr>
<td>0.400 - 0.425</td>
<td>6</td>
</tr>
<tr>
<td>0.425 - 0.450</td>
<td>6</td>
</tr>
<tr>
<td>0.450 - 0.475</td>
<td>6</td>
</tr>
<tr>
<td>0.475 - 0.500</td>
<td>6</td>
</tr>
</tbody>
</table>
### DISTRIBUTION OF SOLUTION DIP RANGES

<table>
<thead>
<tr>
<th>Range</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>8</td>
</tr>
<tr>
<td>5.0</td>
<td>9</td>
</tr>
<tr>
<td>10.0</td>
<td>22</td>
</tr>
<tr>
<td>15.0</td>
<td>35</td>
</tr>
<tr>
<td>20.0</td>
<td>26</td>
</tr>
<tr>
<td>25.0</td>
<td>29</td>
</tr>
<tr>
<td>30.0</td>
<td>24</td>
</tr>
<tr>
<td>35.0</td>
<td>19</td>
</tr>
<tr>
<td>40.0</td>
<td>19</td>
</tr>
<tr>
<td>45.0</td>
<td>17</td>
</tr>
<tr>
<td>50.0</td>
<td>7</td>
</tr>
<tr>
<td>55.0</td>
<td>2</td>
</tr>
<tr>
<td>60.0</td>
<td>0</td>
</tr>
<tr>
<td>65.0</td>
<td>1</td>
</tr>
<tr>
<td>70.0</td>
<td>0</td>
</tr>
<tr>
<td>75.0</td>
<td>0</td>
</tr>
<tr>
<td>80.0</td>
<td>0</td>
</tr>
<tr>
<td>85.0</td>
<td>0</td>
</tr>
<tr>
<td>90.0</td>
<td>0</td>
</tr>
</tbody>
</table>

### DISTRIBUTION OF SOLUTION STRIKE RANGES

<table>
<thead>
<tr>
<th>Range</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>8</td>
</tr>
<tr>
<td>5.0</td>
<td>10</td>
</tr>
<tr>
<td>10.0</td>
<td>65</td>
</tr>
<tr>
<td>15.0</td>
<td>65</td>
</tr>
<tr>
<td>20.0</td>
<td>29</td>
</tr>
<tr>
<td>25.0</td>
<td>16</td>
</tr>
<tr>
<td>30.0</td>
<td>5</td>
</tr>
<tr>
<td>35.0</td>
<td>4</td>
</tr>
<tr>
<td>40.0</td>
<td>3</td>
</tr>
<tr>
<td>45.0</td>
<td>3</td>
</tr>
<tr>
<td>50.0</td>
<td>8</td>
</tr>
<tr>
<td>55.0</td>
<td>1</td>
</tr>
<tr>
<td>60.0</td>
<td>0</td>
</tr>
<tr>
<td>65.0</td>
<td>0</td>
</tr>
<tr>
<td>70.0</td>
<td>0</td>
</tr>
<tr>
<td>75.0</td>
<td>0</td>
</tr>
<tr>
<td>80.0</td>
<td>1</td>
</tr>
<tr>
<td>85.0</td>
<td>0</td>
</tr>
<tr>
<td>90.0</td>
<td>0</td>
</tr>
</tbody>
</table>

### DISTRIBUTION OF SOLUTION RAKE RANGES

<table>
<thead>
<tr>
<th>Range</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>8</td>
</tr>
<tr>
<td>10.0</td>
<td>32</td>
</tr>
<tr>
<td>15.0</td>
<td>40</td>
</tr>
<tr>
<td>20.0</td>
<td>60</td>
</tr>
<tr>
<td>25.0</td>
<td>79</td>
</tr>
<tr>
<td>30.0</td>
<td>23</td>
</tr>
<tr>
<td>35.0</td>
<td>6</td>
</tr>
<tr>
<td>40.0</td>
<td>2</td>
</tr>
</tbody>
</table>

21# events read. 21# processed
B.5 Portion of the FIT FILE for one selected earthquake.

<table>
<thead>
<tr>
<th>SLIP ANGLE = -19°</th>
<th>STRIKE:</th>
<th>115</th>
<th>125</th>
<th>135</th>
<th>145</th>
<th>155</th>
<th>165</th>
<th>175</th>
<th>185</th>
<th>195</th>
<th>205</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>565</td>
<td>558</td>
<td>554</td>
<td>543</td>
<td>524</td>
<td>518</td>
<td>512</td>
<td>506</td>
<td>500</td>
<td>495</td>
<td>490</td>
</tr>
<tr>
<td>10</td>
<td>542</td>
<td>527</td>
<td>515</td>
<td>505</td>
<td>498</td>
<td>494</td>
<td>490</td>
<td>487</td>
<td>483</td>
<td>479</td>
<td>475</td>
</tr>
<tr>
<td>15</td>
<td>519</td>
<td>505</td>
<td>499</td>
<td>494</td>
<td>488</td>
<td>484</td>
<td>480</td>
<td>476</td>
<td>472</td>
<td>468</td>
<td>464</td>
</tr>
<tr>
<td>20</td>
<td>507</td>
<td>496</td>
<td>488</td>
<td>482</td>
<td>477</td>
<td>472</td>
<td>468</td>
<td>464</td>
<td>460</td>
<td>456</td>
<td>452</td>
</tr>
<tr>
<td>25</td>
<td>495</td>
<td>486</td>
<td>478</td>
<td>473</td>
<td>468</td>
<td>464</td>
<td>459</td>
<td>455</td>
<td>451</td>
<td>448</td>
<td>444</td>
</tr>
<tr>
<td>30</td>
<td>485</td>
<td>477</td>
<td>469</td>
<td>465</td>
<td>461</td>
<td>457</td>
<td>454</td>
<td>451</td>
<td>448</td>
<td>445</td>
<td>442</td>
</tr>
<tr>
<td>35</td>
<td>477</td>
<td>470</td>
<td>463</td>
<td>459</td>
<td>456</td>
<td>452</td>
<td>449</td>
<td>446</td>
<td>443</td>
<td>440</td>
<td>437</td>
</tr>
<tr>
<td>40</td>
<td>470</td>
<td>464</td>
<td>458</td>
<td>454</td>
<td>451</td>
<td>448</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
</tr>
<tr>
<td>45</td>
<td>464</td>
<td>459</td>
<td>454</td>
<td>450</td>
<td>447</td>
<td>444</td>
<td>441</td>
<td>438</td>
<td>435</td>
<td>432</td>
<td>429</td>
</tr>
<tr>
<td>50</td>
<td>459</td>
<td>455</td>
<td>451</td>
<td>448</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
</tr>
<tr>
<td>55</td>
<td>455</td>
<td>451</td>
<td>448</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
</tr>
<tr>
<td>60</td>
<td>451</td>
<td>448</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
</tr>
<tr>
<td>65</td>
<td>448</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
</tr>
<tr>
<td>70</td>
<td>445</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
</tr>
<tr>
<td>75</td>
<td>442</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
<td>412</td>
</tr>
<tr>
<td>80</td>
<td>439</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
<td>412</td>
<td>409</td>
</tr>
<tr>
<td>85</td>
<td>436</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
<td>412</td>
<td>409</td>
<td>406</td>
</tr>
<tr>
<td>90</td>
<td>433</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
<td>412</td>
<td>409</td>
<td>406</td>
<td>403</td>
</tr>
<tr>
<td>95</td>
<td>430</td>
<td>427</td>
<td>424</td>
<td>421</td>
<td>418</td>
<td>415</td>
<td>412</td>
<td>409</td>
<td>406</td>
<td>403</td>
<td>400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SLIP ANGLE = -10°</th>
<th>STRIKE:</th>
<th>115</th>
<th>125</th>
<th>135</th>
<th>145</th>
<th>155</th>
<th>165</th>
<th>175</th>
<th>185</th>
<th>195</th>
<th>205</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>561</td>
<td>554</td>
<td>551</td>
<td>543</td>
<td>532</td>
<td>527</td>
<td>522</td>
<td>517</td>
<td>512</td>
<td>508</td>
<td>503</td>
</tr>
<tr>
<td>10</td>
<td>542</td>
<td>532</td>
<td>524</td>
<td>517</td>
<td>511</td>
<td>506</td>
<td>501</td>
<td>497</td>
<td>492</td>
<td>487</td>
<td>482</td>
</tr>
<tr>
<td>15</td>
<td>525</td>
<td>517</td>
<td>511</td>
<td>505</td>
<td>499</td>
<td>494</td>
<td>489</td>
<td>484</td>
<td>479</td>
<td>474</td>
<td>470</td>
</tr>
<tr>
<td>20</td>
<td>512</td>
<td>506</td>
<td>499</td>
<td>494</td>
<td>488</td>
<td>483</td>
<td>478</td>
<td>473</td>
<td>468</td>
<td>463</td>
<td>459</td>
</tr>
<tr>
<td>25</td>
<td>501</td>
<td>496</td>
<td>489</td>
<td>484</td>
<td>478</td>
<td>473</td>
<td>468</td>
<td>463</td>
<td>458</td>
<td>453</td>
<td>449</td>
</tr>
<tr>
<td>30</td>
<td>491</td>
<td>486</td>
<td>481</td>
<td>476</td>
<td>471</td>
<td>466</td>
<td>461</td>
<td>456</td>
<td>451</td>
<td>447</td>
<td>443</td>
</tr>
<tr>
<td>35</td>
<td>483</td>
<td>478</td>
<td>473</td>
<td>468</td>
<td>463</td>
<td>458</td>
<td>453</td>
<td>448</td>
<td>443</td>
<td>439</td>
<td>435</td>
</tr>
<tr>
<td>40</td>
<td>476</td>
<td>471</td>
<td>466</td>
<td>461</td>
<td>456</td>
<td>451</td>
<td>446</td>
<td>441</td>
<td>436</td>
<td>432</td>
<td>428</td>
</tr>
<tr>
<td>45</td>
<td>471</td>
<td>466</td>
<td>461</td>
<td>456</td>
<td>451</td>
<td>446</td>
<td>441</td>
<td>436</td>
<td>432</td>
<td>428</td>
<td>424</td>
</tr>
<tr>
<td>50</td>
<td>466</td>
<td>461</td>
<td>456</td>
<td>451</td>
<td>446</td>
<td>441</td>
<td>436</td>
<td>432</td>
<td>428</td>
<td>424</td>
<td>420</td>
</tr>
<tr>
<td>55</td>
<td>462</td>
<td>457</td>
<td>452</td>
<td>447</td>
<td>442</td>
<td>437</td>
<td>432</td>
<td>427</td>
<td>423</td>
<td>419</td>
<td>415</td>
</tr>
<tr>
<td>60</td>
<td>458</td>
<td>453</td>
<td>448</td>
<td>443</td>
<td>438</td>
<td>433</td>
<td>428</td>
<td>423</td>
<td>419</td>
<td>415</td>
<td>411</td>
</tr>
<tr>
<td>65</td>
<td>454</td>
<td>449</td>
<td>444</td>
<td>439</td>
<td>434</td>
<td>429</td>
<td>424</td>
<td>419</td>
<td>415</td>
<td>411</td>
<td>407</td>
</tr>
<tr>
<td>70</td>
<td>450</td>
<td>445</td>
<td>440</td>
<td>435</td>
<td>430</td>
<td>425</td>
<td>420</td>
<td>415</td>
<td>411</td>
<td>407</td>
<td>403</td>
</tr>
<tr>
<td>75</td>
<td>446</td>
<td>441</td>
<td>436</td>
<td>431</td>
<td>426</td>
<td>421</td>
<td>416</td>
<td>411</td>
<td>407</td>
<td>403</td>
<td>399</td>
</tr>
<tr>
<td>80</td>
<td>442</td>
<td>437</td>
<td>432</td>
<td>427</td>
<td>422</td>
<td>417</td>
<td>412</td>
<td>407</td>
<td>403</td>
<td>399</td>
<td>395</td>
</tr>
<tr>
<td>85</td>
<td>438</td>
<td>433</td>
<td>428</td>
<td>423</td>
<td>418</td>
<td>413</td>
<td>408</td>
<td>403</td>
<td>399</td>
<td>395</td>
<td>391</td>
</tr>
<tr>
<td>90</td>
<td>434</td>
<td>429</td>
<td>424</td>
<td>419</td>
<td>414</td>
<td>409</td>
<td>404</td>
<td>400</td>
<td>396</td>
<td>392</td>
<td>388</td>
</tr>
<tr>
<td>95</td>
<td>430</td>
<td>425</td>
<td>420</td>
<td>415</td>
<td>410</td>
<td>405</td>
<td>400</td>
<td>396</td>
<td>392</td>
<td>388</td>
<td>384</td>
</tr>
<tr>
<td>SLIP ANGLE = -15°</td>
<td>STRIKE:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>626</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>627</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>655</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>683</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>692</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>672</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>655</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>630</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>603</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>575</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>549</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>523</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>498</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>473</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>428</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>428</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>422</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>404</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>422</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SLIP ANGLE = -14°</th>
<th>STRIKE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIP</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>626</td>
</tr>
<tr>
<td>10</td>
<td>656</td>
</tr>
<tr>
<td>15</td>
<td>686</td>
</tr>
<tr>
<td>20</td>
<td>632</td>
</tr>
<tr>
<td>25</td>
<td>692</td>
</tr>
<tr>
<td>30</td>
<td>672</td>
</tr>
<tr>
<td>35</td>
<td>655</td>
</tr>
<tr>
<td>40</td>
<td>630</td>
</tr>
<tr>
<td>45</td>
<td>603</td>
</tr>
<tr>
<td>50</td>
<td>575</td>
</tr>
<tr>
<td>55</td>
<td>549</td>
</tr>
<tr>
<td>60</td>
<td>523</td>
</tr>
<tr>
<td>65</td>
<td>498</td>
</tr>
<tr>
<td>70</td>
<td>473</td>
</tr>
<tr>
<td>75</td>
<td>428</td>
</tr>
<tr>
<td>80</td>
<td>428</td>
</tr>
<tr>
<td>85</td>
<td>422</td>
</tr>
<tr>
<td>90</td>
<td>404</td>
</tr>
<tr>
<td>95</td>
<td>422</td>
</tr>
<tr>
<td>DIP</td>
<td>SLIP ANGLE = -13°</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
</tr>
<tr>
<td>5</td>
<td>561 641 616 6# 583 566 556 55# 542 534 529 526 522 524 519 504 495 486</td>
</tr>
<tr>
<td>10</td>
<td>697 665 629 6# 574 548 525 508 492 477 463 458 439 423 414 396 394 392</td>
</tr>
<tr>
<td>15</td>
<td>762 728 675 621 557 51# 454 408 365 325 287 284 276 301 309 315 311 315 321</td>
</tr>
<tr>
<td>20</td>
<td>755 696 631 527 442 387 33# 273 211 177 176 177 187 202 217 228 240 254</td>
</tr>
<tr>
<td>25</td>
<td>714 642 563 467 395 32# 242 176 159 147 142 138 148 155 168 187 204 218 234</td>
</tr>
<tr>
<td>30</td>
<td>677 619 521 429 343 273 196 149 131 123 123 132 142 146 164 182 204 223</td>
</tr>
<tr>
<td>35</td>
<td>647 576 485 395 308 236 159 127 128 119 116 109 114 127 133 146 160 192 216</td>
</tr>
<tr>
<td>40</td>
<td>624 525 459 37# 287 19@ 180 166 99 92* 96* 92* 96* 10# 119 128 147 163 224</td>
</tr>
<tr>
<td>45</td>
<td>57# 69# #75 66# 54# 17# 18# 19# 20# 21# 22# 23# 24# 25# 26# 27# 28# 29# 30#</td>
</tr>
<tr>
<td>55</td>
<td>500 449 373 30# 213 130 114 106 97* 84* 70* 87* 89* 88* 111 130 168 194 235</td>
</tr>
<tr>
<td>60</td>
<td>491 435 362 296 284 130 119 103 92* 75* 73* 84* 84* 88* 112 143 177 202 245</td>
</tr>
<tr>
<td>65</td>
<td>486 422 352 28# 194 137 128 99 92* 74* 71* 79* 79* 89* 113 150 186 219 256</td>
</tr>
<tr>
<td>70</td>
<td>47# 410 333 27# 195 144 128 97* 75* 71* 67* 79* 90* 120 178 203 234 270</td>
</tr>
<tr>
<td>75</td>
<td>464 400 327 256 194 144 123 97* 72* 67* 55* 66* 88* 93* 148 180 223 252 286</td>
</tr>
<tr>
<td>80</td>
<td>459 396 329 242 180 152 111 91* 74* 56* 54* 65* 82* 117 162 211 243 283 330</td>
</tr>
<tr>
<td>90</td>
<td>462 406 347 245 176 141 117 88* 64* 60* 60* 60* 119 176 213 275 333 392 465</td>
</tr>
<tr>
<td>95</td>
<td>460 404 338 239 188 174 128 1#5 84* 78* 82* 108 161 2#5 257 331 393 474 556</td>
</tr>
</tbody>
</table>
B.6 Graphic output from program FPLOT for selected earthquakes.
Note the multiple solutions for event 840909.
Note the low value of STDR for event 750426.
GEYSERS, CALIFORNIA EARTHQUAKES

840909 2 8 37.03 38-47.05 122.46.38 1.32 2.54 17 51 2.0 0.03 0.1 0.1 A
85 80-170 0.03 17 4.68 0.58 0.00 20 20 30 B-B

DISCREPANT OBSERVATIONS
GRL 8.3 164 50 IPA1
NMH 26.3 170 54 IPA1
GEYSERS, CALIFORNIA EARTHQUAKES

840909  28 37.03 30-47.05 122-46.38  1.32  2.54 17  51  2.0  0.03  0.1  0.1 A
130  50-100  0.06  17  4.68  0.50  0.00  15  10  10 B-A

DISSAPPOINT OBSERVATIONS
SIRI. DIST. BRT. DIN. PARK
GRAX  0.3  169  60  140
GEYSERS, CALIFORNIA EARTHQUAKES

840909 28 37.03 38-47.05 122-46.38 1.32 2.54 17 51 2.0 0.03 0.1 0.1
230 50 90 0.00 17 4.68 0.42 0.00 10 0 10 A-A

△ DILATION
○ COMPRESSION
GEYSERS, CALIFORNIA EARTHQUAKES

840410 4 2 2.49 30-48.52 122-47.63 1.38 2.95 25 38 0.0 0.05 0.1 0.1 A
75 05-140 0.02 25 4.09 0.56 0.00 10 15 20 A-A

DISCREPANT OBSERVATIONS
S|A|O|T|D|O|I|S|T|O|D|I|S|T|O|N|S|O|N|S|O|N
GEX 0.1 177 175 175 175 175

Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Δ Δ Δ Δ Δ Δ
Graphic output from program FPPAGE for the first 42 solutions listed in section B.2.
APPENDIX C

C.1 FPFIT FORTRAN listing

C.2 FPPLT FORTRAN listing

C.3 FPPAGE FORTRAN listing
PROGRAM FPFIT

VERSION 1.0 - OCTOBER 31, 1985

PURPOSE: CALCULATE DOUBLE-COUPLE FAULT PLANE SOLUTIONS FROM P-WAVE FIRST MOTIONS (SEE REASENBERG, P. AND D. OPPENHEIMER, FPFIT, FPPLOT AND FPPAGE FORTRAN COMPUTER PROGRAMS FOR CALCULATING AND DISPLAYING EARTHQUAKE FAULT-PLANE SOLUTIONS, U.S. GEOLOGICAL SURVEY OPEN-FILE REPORT 85-???).


2. A PARAMETER CONTROL FILE WHICH IS READ ON LOGICAL UNIT CUNIT (=-1)

REQUIRED ROUTINES: ALL ROUTINES ARE ENCLOSED

DEPARTURES FROM FORTRAN-77 STANDARD:
1. KEYWORDS "READONLY" AND "CARRIAGECONTROL = LIST" IN OPEN STATEMENTS
2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS

OUTPUT: 1. AN ASCII FILE OF HYP07I SUMMARY CARDS EXTENDED WITH FAULT PLANE SOLUTION PARAMETERS ON LOGICAL UNIT SUNIT (=-4). THIS FILE SERVES AS INPUT TO PROGRAMS "GLOT" AND "PLOTPT".

2. AN ASCII FILE CONSISTING, FOR EACH EARTHQUAKE, OF THE HYP07I EXTENDED SUMMARY CARD, FOLLOWED BY NEIGHBORING SOLUTIONS (WITHIN 90X CONFIDENCE LIMITS), FOLLOWED BY INDIVIDUAL P-PHASE INFORMATION, ON LOGICAL UNIT PUNIT (=-3). THIS FILE SERVES AS INPUT TO PROGRAMS "PLOTFM" AND "FPPAGE".

3. AN OPTIONAL ASCII FILE OF THE MINIMIZED FIT FUNCTION ABOUT THE BEST SOLUTION ON LOGICAL UNIT FUNIT (=-9)

4. AN ASCII FILE DESCRIBING ANY ERRORS IN THE CONTROL FILE, HYP07I FILE, PRESENCE OF MULTIPLE MECHANISMS, A SUMMARY OF POLARITY DISCREPANCIES BY STATION AND READING QUALITY, AND THE DISTRIBUTION OF STRIKE, DIP, AND RAKE UNCERTAINTIES ON LOGICAL UNIT EUNIT (=-8)

AUTHORS: PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK. SOME OF THE ROUTINES WERE ADAPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.

MARK MATTHEWS, STANFORD UNIVERSITY, PROVIDED ASSISTANCE IN THE ERROR PROPAGATION ANALYSIS.

INTEGER
  CUNIT  LOGICAL UNIT # OF INPUT CONTROL FILE
  EUNIT  LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
  FUNIT  LOGICAL UNIT # OF OUTPUT OF FIT LISTING FOR ALL STRIKES, DIPS
  IUNIT  LOGICAL UNIT # OF HYP07I LISTING FILE (INPUT FILE)
  MXDIP  MAXIMUM # OF DIP INCREMENTS PERMITTED
  MXQUAL  MAXIMUM # OF QUALITIES PERMITTED
  MXRAKE  MAXIMUM # OF RAKE INCREMENTS PERMITTED
  MXSLNS  MAXIMUM # OF MULTIPLE SOLUTIONS PERMITTED
  MXSTAT  MAXIMUM # OF STATIONS PERMITTED
  MXSTRK  MAXIMUM # OF STRIKE INCREMENTS PERMITTED
  PUNIT  LOGICAL UNIT # OF OUTPUT OF EXTENDED SUMMARY AND RAY PARAMETERS
  SUNIT  LOGICAL UNIT # OF OUTPUT OF EXTENDED SUMMARY CARDS

PARAMETER (CUNIT = 1, EUNIT = 8, FUNIT = 9, IUNIT = 2, MXDIP = 19,
  & MXQUAL = 8, MXRAKE = 1B, MXSLNS = 2B, MXSTAT = 1BB, MXSTRK = 19,
  & PUNIT = 3, SUNIT = 4)

REAL
  AERR  ALLOWABLE DIFFERENCE BETWEEN CORRESPONDING ANGLES OF COMPLIMENTARY PLANES
  AIN(MXSTAT)  RAY INCIDENCE ANGLES IN DEGREES
  AINR  AIN CONVERTED TO RADIANS
  AZ(MXSTAT)  RAY AZIMUTH ANGLES (CORRESPONDING TO AIN)
  AVWT  MEAN OBSERVATIONAL WEIGHT OF DATA USED IN SOLUITION
  AZR  AZ CONVERTED TO RADIANS
  BDFLAG  SIGNALS POLARITY DISCREPANCY WITH BEST FIT SOLUTION
  BEST  FLAG: TRUE-BEST SOLUTION FOR EVENT
REAL, COMPL, LOGICAL, REAL

REAL BOT(MXDIP, MXSTRK, MXRAKE)  SUM OF PRODUCT OF OBSERVED AND PREDICTED WEIGHTS
REAL COEF(MXSTAT, 6)  COEFFICIENTS BY WHICH MOMENT TENSOR MULTIPLIED TO GIVE P RADIATION PATTERN
REAL DD2  DIP ANGLE OF AUXILIARY SOLUTION
REAL DDEL  DIP DIRECTION OF AUXILIARY SOLUTION
REAL DELF  FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DELG  FAULT DIP INCREMENT IN DEGREES FOR FINE SEARCH
REAL DELI(S)  FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
REAL DELIF(S)  FAULT DIP ANGLE IN DEGREES FOR FINE SEARCH
REAL DIFITC  INCREMENT TO COARSE FIT FUNCTION
REAL DIP  DIP ANGLE OF BEST SOLUTION
REAL DIST(MXSTAT)  EPICENTRAL DISTANCE
REAL DISTMX  MAXIMUM PERMITTED EPICENTRAL DISTANCE
REAL DLAMC  FAULT RAKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DLAMF  FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL DPHIC  FAULT STRIKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DPHIF  FAULT STRIKE IN DEGREES FOR FINE SEARCH
REAL ERATE(MXQUAL)  ASSUMED WEIGHTED ERROR RATES FOR EACH DATA CLASS

CHARACTER*80 DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
CHARACTER*52 DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
LOGICAL FIRST  SOLUTION FIT: WEIGHTED MEASURE OF AGREEMENT BETWEEN OBS, PRED POLARITIES
REAL FIT(MXDIP, MXSTRK, MXRAKE)  SOLUTION FIT: WEIGHTED MEASURE OF AGREEMENT BETWEEN OBS, PRED POLARITIES
REAL FITLIM  UPPER BOUND ON "GOOD" SOLUTIONS: 5% ABOVE FITMIN
REAL FITMIN  FIT OF BEST SOLUTION CORRESPONDING TO FIT(J1, M1, M1)
CHARACTER*1 FITLIM THEN "", OTHERWISE BLANK
REAL FMAGMN  OUTPUT STRING: IF FIT < FITLIM THEN "", OTHERWISE BLANK
LOGICAL FPIMP  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
CHARACTER(3) FITQUAL  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
REAL GFIT(MXDIP, MXSTRK, MXRAKE)  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER I  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER ID  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IDIP  DIP ANGLE OF NEARBY SOLUTIONS
INTEGER IDIPI  DIP ANGLE OF BEST FIT
INTEGER IDIPDR  DIP DIRECTION OF NEARBY SOLUTIONS
INTEGER IND(MXSLNS, 3)  DIP INDEX OF BEST SOLUTION
INTEGER INDPI  DIP DIRECTION OF BEST FIT
INTEGER IDRNG  DIP DIRECTION OF NEARBY SOLUTIONS
INTEGER IERR  OUTPUT STRING: IF FIT < FITLIM THEN "", OTHERWISE BLANK
INTEGER IEVP  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IEVR  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IFIT(MXSTRK)  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IGOOD(MXDIP=MXSTRK=MXRAKE, 4)  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IFEFF  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IND(MXSTAT)  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER INDEX  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IPWNT  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IPWT  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IRES  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER IRRNG  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER ISLIP  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER ISLIP1  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER ISRNG  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER J  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER J1  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER JMAX  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER JMIN  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER K  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
CHARACTER*4 KILSTA(MXSTAT)  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
INTEGER L  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
CHARACTER*122 LINE  DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
LOOP INDEX OVER RAKE

RAKE INDEX OF BEST SOLUTION

MINIMUM NUMBER OF OBSERVATIONS REQUIRED

LARGEST RAKE INDEX OF SOLUTION WITH FIT <FITLIM ABOUT (J1, N1, M1)

SMALLEST RAKE INDEX OF SOLUTION WITH FIT <FITLIM ABOUT (J1, N1, M1)

LOOP INDEX OF STRIKE

STRIKE INDEX OF BEST SOLUTION

NUMBER OF FAULT DIP INCREMENTS FOR COARSE SEARCH

NUMBER OF FAULT DIP INCREMENTS FOR FINE SEARCH

FAULT NUMBER OF DIP INCREMENTS FOR FINE SEARCH

NUMBER OF DIP SOLUTION RANGES BINNED INTO DELF DEGREE INCREMENTS

NUMBER OF DISTINCT SOLUTIONS FOUND BY HHOG

NUMBER OF EVENTS TO PROCESS

NUMBER OF SOLUTIONS BINNED INTO .25 FIT INCREMENTS

NUMBER OF "GOOD" SOLUTIONS FOUND IN COARSE SEARCH

NUMBER OF FAULT RAKE INCREMENTS FOR COARSE SEARCH

NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH

DEFAULT NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH

NUMBER OF FAULT STRIKE INCREMENTS FOR COARSE SEARCH

NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH

DEFAULT NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH

-1-EOF, 0-SKIP EVENT, NR>0 -> NUMBER OF STATIONS

NUMBER OF REVERSED STATIONS

NUMBER OF RAKE SOLUTION RANGES BINNED INTO DLAM DEGREE INCREMENTS

NUMBER OF PLANES STORED IN ARRAY SOLNS

NUMBER OF SOLUTIONS BINNED INTO DPHIF DEGREE INCREMENTS

NUMBER OF SOLUTIONS HAVING FIT WITHIN 5X OF FITMIN

TOTAL # OF STATIONS REPORTING FOR ENTIRE DATA SET

NUMBER OF REVERSED STATIONS

RADIATION AMPLITUDE CORRESPONDING AIN, AZ.

INITIAL FAULT STRIKE ANGLE IN DEGREES FOR FINE SEARCH

TERMINATING FAULT STRIKE ANGLE IN DEGREES FOR FINE SEARCH

OBSERVED FIRST MOTION POLARITIES; .5-COMPRESSION, -.5-DILATATION

RADIATION AMPLITUDE CORRESPONDING AIN, AZ.

COMPUTED STANDARD DEVIATION OF FIT BASED ON DATA ERRORS

SLIP ANGLE OF BEST SOLUTION

ARRAY OF FINAL SOLUTIONS USED TO CHECK FOR REDUNDANCY

NAMES OF ALL STATIONS REPORTING

STATION DISTRIBUTION RATIO

STATION NAMES PER EVENT

SCRATCH VARIABLE

STATION NAMES

STATION NAMES

MORE TENSORS IN UPPER TRIANGULAR SYMMETRIC STORAGE MODE

CARTESIAN P-WAVE DIRECTION VECTOR (POSITIVE UP, SOUTH, EAST)

WEIGHTS ASSOCIATED WITH QUALITIES

MAXIMUM WEIGHT
REAL WITORS(MXSTAT)  I OBSERVED FIRST MOTIONS WEIGHTS
REAL XLAM1(MXRAKE)  I FAULT RAKE ANGLE IN DEGREES
REAL XLAMC(MXRAKE)  I FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
REAL XLAMC0  I INITIAL FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
REAL XLAMIF  I TERMINATING FAULT RAKE ANGLE IN DEGREES FOR FINE SEARCH

C INITIALIZE STATISTICS ARRAYS TO ZERO

DATA NSTAT, NFIT /0, 20*/
DATA QCNT, QCNTWT /MXQUAL*0, MXQUAL*0, MXQUAL*0.0, MXQUAL*0.0/
DATA NDRNG, NSRNG, NRRNG /MXDIP*0, MXSTRK*0, MXRAKE*0/

OPEN FILES

OPEN (UNIT - SUNIT, STATUS - 'NEW', FORM - 'FORMATTED', ACCESS
C & - 'SEQUENTIAL', CARRIAGECONTROL - 'LIST')
OPEN (UNIT - PUNIT, STATUS - 'NEW', FORM - 'FORMATTED', ACCESS
C & - 'SEQUENTIAL')
OPEN (UNIT - EUNIT, STATUS - 'NEW', FORM - 'FORMATTED', ACCESS
C & - 'SEQUENTIAL', READONLY, IOSTAT = IERR)
IF (IERR .NE. 0) THEN
PRINT *, '***** FPFIT: ERROR OPENING HYPO FILE; ERROR CODE = ',
& IERR, ', LOGICAL UNIT # = ', IUNIT, ' *****
STOP
END IF

C READ IN CONTROL PARAMETERS
C
IF (.NOT. FPINP (CUNIT, DDEL, DDELRF, DDEL#C, DIFTC, DISTMX,
& DLAMC, DLAMF, DPIC, DPICF, EFASE, EUNIT, FMAGMN, IJEFF, IPRNT,
& IRES, IUNIT, KILST, MINOBS, MKDIP, MKQUAL, MXRAKE, MXSTAT,
& MXSTRK, NDELC, NDELDF, NEV, NKL, NLAMC, NLAMDF, NPIC,
& NPHICF, NREV, PHIC, REVSTA, WEIGHT, XLAMC0)) STOP
IF (IPRNT .EQ. I) OPEN (UNIT - FUNIT, STATUS - 'NEW', FORM -
& 'FORMATTED', ACCESS - 'SEQUENTIAL')

WRITE (SUNIT, 11) ' DATE ORIGIN LAT N LONG W DEPTH
1 IMAG NO GAP DM RMS ERH ERZ QL DD DA SA FIT NR SMWT BOT
2 XX S D R RNG O'
11 FORMAT(A)

C FIND MAXIMUM WEIGHT
C
WTMAX = 0.
DO 15 I = 1, MXQUAL
IF (WEIGHT(I) .GT. WTMAX) WTMAX = WEIGHT(I)
CONTINUE
READ NEXT EVENT

IEVR = IEVR + 1
CALL READEQ (AIN, AZ, DIST, DISTMX, EUNIT, EVENT, FMAGMN, & IJEFF, IUNIT, KILSTA, MINOBS, MXQUAL, MXSTAT, MXIL, NR, NREV, & PFOBS, PRCTMX, PRMK, REVSTA, SIGMAF, STN, SUMWT, WEIGHT, WTOBS)
IF (NR .EQ. -1 .OR. IEVP .EQ. NEV) THEN
END OF DATA, CLOSE FILES
IEVR = IEVR - 1
CLOSE (CUNIT)
CLOSE (IUNIT)
CLOSE (PUNIT)
CLOSE (SUNIT)
IF (IPRNT .EQ. 1) CLOSE (FUNIT)
IF (NSTAT .GT. 0) THEN
CALL FPOUT (DDELF, DLAMF, OPHIF, ERATE, EUNIT, IEVP, IEVR, & IMO, IRES, MXDIP, MXQUAL, MXRAKE, MXSTAT, MXSTRK, NOELF, NDRNG, & NFIT, NLAMF, NPHIF, NREV, NRRNG, NSRNG, NSTAT, QCNT, QCNTWT, & REVSTA, SCNT, SCNTWT, STAT)
ELSE
WRITE (EUNIT, *) '****** FPFIT ERROR: NO EVENTS SATISFY INPUT & CRITERIA ****** '
END IF
CLOSE (EUNIT)
STOP
END IF

INSUFFICIENT READINGS SKIP EVENT
IF (NR .EQ. 0) GOTO 28
GOOD EVENT: BEGIN EVENT LOOP
SET UP P-WAVE DIRECTION UNIT VECTOR (UP, SOUTH, EAST) FOR EACH RAY
DO 30 I = 1, NR
AINR = AIN(I)*RAD
AZR = AZ(I)*RAD
U(1) = -COS(AINR)
U(2) = -SIN(AINR)*COS(AZR)
U(3) = SIN(AINR)*SIN(AZR)
FIND EXCITATION COEFFICIENTS FOR DETERMINING FAR-FIELD P RADIATION PATTERN
CALL PEXCF (COEF, I, MXSTAT, U)
CONTINUE
COARSE LOOP THROUGH FAULT MODELS
IEVP = IEVP + 1
FIRST = .TRUE.
CALL SEARCH (BOT, COEF, DDELC, DEL, DELC, DELRC, DFITC, DLAMC, & DPHIC, FIRST, FIT, FITMIN, FLAG, GFIT, IGOOD, IPRNT, JI, MI, & MXDIP, MXRAKE, MXSTAT, MXSTRK, NI, NOELC, NG, NLAMC, NPHIC, NR, & NSTAT, PHI, PHIC, PHIRC, PFOBS, RAD, WTOBS, XLM, XLMC, XLAMC)
Determine distinct solutions from array of "good" solutions in coarse search
CALL HHOG (EUNIT, 01, N1, M1, IGOOD, GFIT, NG, IDST, NSTAT, & MXDIP, MXSLNS, MXSTRK, MXRAKE, NOELC, NPHIC, NLAMC)
C BEGIN LOOP FOR FINE SEARCH CENTERED ON EACH DISTINCT SOLUTION FOUND BY HHOG

C

FIT90 - 1.282 * SIGMAF
N'OL = 0
AIH 2.-AMAX1(DDLF, DLAMF, DPHIF)
IDQ -1. ID = 1, ND, T
JI = IDST(I, 1)
NI = IDST(J, 2)
MI = IDST(I, 3)
REST = .FALSE.
IF (ID.FEQ.1) BEST = .TRUE.
FIRST = .FALSE.

C DETERMINE FINE SEARCH RANGE FOR UNRESTRICTED SEARCH

C

IF (IRES.EQ.0) THEN
NDCLF = NDELFDF
NPHIF = NPHIFDF
NLAMF = NLAMFD
DEL0F = DELC(JI) - FLOAT(NDELF/2)*DDELF
PHI0F = PHIC(NI) - FLOAT(NPHIF/2)*DPHIF
XLAM0F = XLAMC(MI) - FLOAT(NLAMF/2)*DLAMF
ELSE
C
C DETERMINE FINE SEARCH DIP RANGE FOR RESTRICTED SEARCH

C

DEL0F = DELC(JI) - FLOAT(NDELC/2)*DDEL0C
IF (DEL0F.LT. DELC(JI)) DEL0F = DELC(JI)
DEL0F = DELC(JI) - FLOAT(NDELC/2)*DDEL0C
IF (DEL0F.GT. DELC(JI)) DEL0F = DELC(JI)
ELSE
C
C DETERMINE FINE SEARCH STRIKE RANGE FOR RESTRICTED SEARCH

C

PHI0F = PHIC(NI) - FLOAT(NPHIC/2)*DPHIC
IF (PHI0F.LT. PHIC(NI)) PHI0F = PHIC(NI)
PHI0F = PHIC(NI) - FLOAT(NPHIC/2)*DPHIC
IF (PHI0F.GT. PHIC(NI)) PHI0F = PHIC(NI)
NPHIF = INT(PHI0F/DPHIC) + 1
IF (NPHIF.GT. MXPHIF) THEN
NPHIF = MXPHIF
PHI0F = PHIC(NI) - FLOAT(NPHIF/2)*DPHIC
IF (PHI0F.LT. PHIC(NI)) PHI0F = PHIC(NI)
ELSE
C
C DETERMINE FINE SEARCH RAKE RANGE FOR RESTRICTED SEARCH

C

XLAM0F = XLAMC(MI) - FLOAT(NLAMC/2)*DLAMC
IF (XLAM0F.LT. XLAMC(MI)) XLAM0F = XLAMC(MI)
XLAM0F = XLAMC(MI) - FLOAT(NLAMC/2)*DLAMC
IF (XLAM0F.LT. XLAMC(NLAMC)) XLAM0F = XLAMC(NLAMC)
NLAGF = INT((XLAGF - XLAGF)/DLAMF) + 1
IF (NLAGF .GT. MXRAKE) THEN
  NLAGF = MXRAKE
  XLAGF = XLMC(MI) - FLOAT(NLAGF)/DLAMF
END IF
IF (XLAGF .LT. XLMC(MI)) THEN
  NLAGF = NLAGF - 1
  GOTO 68
END IF
END IF
C
C DO FINE SEARCH
C
CALL SEARCH (RDOT, COEF, DEL, DELC, DELOF, FITQ, DLAMF, & DPHIF, FIRST, FIT, FITMIN, FLAG, GFIT, IDIOGOOD, IPRINT, J1, MI, & MXDIP, MXRAKE, MXSTAT, MXSTRK, NI, NDELF, NG, NLAMF, NPHIF, NR, & MSTAK, PHI, PHIC, PHIF, POBS, RAD, WIOBS, XLM, XLMC, XLAGF)
C
EXPRESS SOLUTION IN TERMS OF DIP DIRECTION, DIP ANGLE, AND SLIP ANGLE.
C
CALL REFMT(J1, IDIP, IDIPDR, ISLIP, PHI(MI), XLM(MI))
C
REPLACE EACH PLANE WITH IDIP = 0 BY ITS AUXILIARY PLANE
C
IF (IDIP .EQ. 0) THEN
  CALL AUXPLN (FLOAT(IDIPDR), IDIP, ISLIP, DD2, DA2, SA2)
END IF
C
FOR CASES WHERE PLANE IS VERTICAL AND RAKE IS -90, FLIP REPRESENTATION TO
C ONE WITH POSITIVE RAKE
C
IF (IDIP .EQ. 90 .AND. ISLIP .EQ. -90) THEN
  ISLIP = ISLIP + 180
  IDIPDR = MOD(IDIPDR + 180, 360)
END IF
C
SKIP SOLUTION IF IT IS COMPLEMENTARY (AUXILIARY PLANE) TO ANOTHER SOLUTION
C OR IF IT IS THE SAME AS ANOTHER SOLUTION IN THE LIST
C
IF (NSOL .EQ. 0 .OR. (NOT. COMPL(SIGNS, NSOL, FLOAT(IDIPDR)), & FLOAT(IDIP), FLOAT(ISLIP), AERR, MXSIGNS)) THEN
  NSOL = NSOL + 1
  SIGNS(NSOL, 1) = IDIPDR
  SIGNS(NSOL, 2) = IDIP
  SIGNS(NSOL, 3) = ISLIP
END IF
C
FIND THE RANGE OF DIP, STRIKE AND RAKE SPANNING EACH GOOD SOLUTION FOR WHICH THE FIT IS .LE. FITLIM
C
FITLIM = FITMIN + FITQ
JMAX = B
JMIN = NDELF
MMAX = B
MMIN = NPHIF
NMIN = NLAGF
DO 78 M = 1, NLAGF
  IF (FIT(J1, N1, M) .LE. FITLIM) THEN
    IF (M .LT. MMIN) MMIN = M
    IF (M .GT. MMAX) MMAX = M
  END IF
  . . .
DO 80 N = I, NPHIF
    IF (FIT(J, N, MI) .LE. FITLIM) THEN
        IF (N .LT. NMIN) NMIN = N
        IF (N .GT. NMAX) NMAX = N
    END IF
CONTINUE
DO 80 J = 1, NDELF
    IF (FIT(J, N, MI) .LE. FITLIM) THEN
        IF (J .LT. JMIN) JMIN = J
        IF (J .GT. JMAX) JMAX = J
    END IF
CONTINUE
C UPDATE DISTRIBUTION OF DIP, STRIKE, RAKE RANGES FOR BEST SOLUTIONS
C
IDRNG = MAX(JMAX - J1, J1 - JMIN)*IFIX(DDELFI)
ISRNG = MAX(MMAX - MI, MI - MM1N)*IFIX(DPHIF)
IRRNG = MAX(MMAX - MI, MI - MM1N)*IFIX(DLAMF)
C
ACCUMULATE STATISTICS ON DISTRIBUTION OF DIP, STRIKE, RAKE RANGES
C FOR BEST SOLUTIONS ONLY
C
IF (BEST .AND. (IRES .EQ. 0)) THEN
    INDEX = IDRNG/IFIX(DDELFI) + 1
    NDRNG(INDEX) = NDRNG(INDEX) + 1
    INDEX = ISRNG/IFIX(DPHIF) + 1
    NSRNG(INDEX) = NSRNG(INDEX) + 1
    INDEX = IRRNG/IFIX(DLAMF) + 1
    NRRNG(INDEX) = NRRNG(INDEX) + 1
END IF
C
ASSIGN QUALITY CODE TO SOLUTION
C
FTOUAL(2:2) = 'I'
    IF (FITMIN .LE. .025) THEN
        FTOUAL(1:1) = 'A'
    ELSE IF (FITMIN .GT. .025) THEN
        FTOUAL(1:1) = 'C'
    ELSE
        FTOUAL(1:1) = 'B'
    END IF

& THEN
    FTOUAL(3:3) = 'A'
    ELSE IF (ISRNG .GT. 40) THEN
        FTOUAL(3:3) = 'C'
    ELSE
        FTOUAL(3:3) = 'B'
    END IF
C
WRITE OUT RESULTS
C
SFLAG = ' '
    IF (.NOT. BEST) SFLAG = '*'
    TEMP = EVENT(1:53)/EVENT(57:60)/EVENT(54:56)/'.'
    EVENT(53:80)
    AVWT = SUMWT/FLOAT(NR)
    STOR = BOT(J1, N1, MI)/SUMWT
    WRITE (EVFIT, 100) IDPDR1, IDIP1, ISLIP1, FIT(J1, N1, MI),
        NR, AVWT, STOR, PRCTX, ISRNG, IRRNG, FTOUAL, SFLAG
100
    FORMAT (14, 13, 14, 2X, F4.2, 13X, 13, 2F5.2, 13X, F4.2,
        13X, 313, 1X, A3, A1)
IF (IPRNT .EQ. 1) WRITE (FUNIT, 110) TEMP, EVFIT  
   FORMAT ('1 *, A80. A52)  
WRITE (SUNIT, 120) TEMP, EVFIT  
   FORMAT (A80, A52)  
WRITE (PUNIT, 130) TEMP, EVFIT  
   FORMAT ('I, A80, A52)  
WRITE (PUNIT, *) NSTAR  
C  
C LOOP OVER SEARCH AREA TO PRINT OUT FIT PARAMETERS  
C  
NSTAR = 0  
DO 200 M = 1, NLAMF  
   IF (IPRNT .EQ. 1) THEN  
      WRITE (FUNIT, 140) IFIX(XLAM(M))  
      FORMAT ('0', //, 4X, 'SLIP ANGLE - *, 14)  
      WRITE (FUNIT, 150) (IFIX(PHI(N )), N = I, NPHIF)  
      FORMAT (' ', ' STRIKE* 1 , 24(2X, 13))  
      WRITE (FUNIT, *) ' DIP*  
   END IF  
   DO 190 J = 1, NDELF  
      DO 170 N = 1, NPHIF  
         IF (IPRNT .EQ. 1) THEN  
            IFIT(N) = IFIX(1000.*FIT(J, N, M))  
            IF (IFIT(N) .EQ. 999) IFIT(N) = 999  
            END IF  
            IF (FLAG(0, N, M) .EQ. '*') THEN  
               NSTAR = NSTAR * 1  
               CALL REFRMT(DEL(J), IDIP, IDIPDR, ISLIP, XLAM(M))  
               CALL SHRFLT(STRIKE, DIP, SLIP, TM)  
         END IF  
      END DO 170  
      CONTINUE  
   END DO 190  
   IF (NSTAR .EQ. 1) THEN  
      LINE = STRING  
   ELSE  
      LINE = LINEUiiNSTAR - 1 )«11 )//STR ING  
   END IF  
   IF (NSTAR .NE. 11) THEN  
      NSTAR = 0  
      WRITE (PUNIT, 160) LINE  
      FORMAT (' ', A)  
   END IF  
END IF  
END IF  
CONTINUE  
IF (IPRNT .EQ. 1) WRITE (FUNIT, 180) IFIX(DEL(J)), & (IFIT(N), FLAG(J, N, M), N = I, NPHIF)  
   FORMAT (' ', 'I4, 14, 6X, 24(13, Al, IX), //)  
180 CONTINUE  
190 CONTINUE  
200 CONTINUE  
IF (NSTAR .NE. 0) WRITE (FUNIT, 160) LINE  
C  
C ACCUMULATE STATISTICS ON FIT PARAMETER DISTRIBUTION FOR BEST SOLUTIONS ONLY  
C  
IF (BEST) THEN  
   INDEX = IFIX(FIT(J1, N1, M1)/.025) * 1  
   IF (INDEX .LT. 20) INDEX = 20  
   NFIT(INDEX) = NFIT(INDEX) + 1  
END IF  
C  
C RECOMPUTE MOMENT TENSOR REPRESENTATION FOR BEST SOLUTION TO CHECK FOR POLARITY DISCREPANCIES  
C  
STRIKE = PHI(N1)*RAD  
DIP = DEL(J1)*RAD  
SLIP = XLAM(M1)*RAD  
CALL SHRFLT (STRIKE, DIP, SLIP, TM)  
DO 260 K = 1, NR
IF (NSTAT .GE. 1) THEN
   DO 210 I = 1, NSTAT
      IF (STN(I) .EQ. STATU)) GOTO 220

210 CONTINUE
   END IF
   NSTAT = NSTAT + 1
   IF (NSTAT .GT. MXSTAT) THEN
      WRITE (EUNIT, *) '**** FPFIT ERROR: # OF STATIONS HAVE
&POLARITY DISCEPANCIES EXCEEDS ', MXSTAT, ' *** *
      WRITE (EUNIT, *) ' STATION ', STN(K), ' NOT REPORTED IN
&FINAL SUMMARY'
   GOTO 240
   END IF
   I = NSTAT
   STAT(NSTAT) = STN(K)
   SCNT(NSTAT, 1) = 0
   SCNT(NSTAT, 2) = 0
   SCNTWT(NSTAT, 1) = 0.
   SCNTWT(NSTAT, 2) = 0.

220 READ (PRMK(K), '(3X, I1)') IPWT
   IF (PRMK(K)(1:1) .EQ. 'X' .OR. PRMK(K)(1:1) .EQ. 'Y')
      & IPWT = IPWT * MXQUAL/2
   C RECOMPUTE RADIATION PATTERN
   C
   PRAD = 0
   DO 230 L = 1, 6
      PRAD = PRAD * TM(L)*COEF(K, L)
   230 CONTINUE
   C CHECK POLARITY AND UPDATE APPROPRIATE STATION COUNT
   C
   IF (SIGN(SIG, PRAD) .NE. POBS(K)) THEN
      IF (BEST) THEN
         SCNT(I, 1) = SCNT(I, 1) + 1
         SCNTWT(I, 1) = SCNTWT(I, 1) + & WTOBS(K)*SORT(ABS(PRAD))
         QCNT(IPWT + 1, 1) = QCNT(IPWT + 1, 1) + 1
         QCNTWT(IPWT + 1, 1) = QCNTWT(IPWT + 1, 1) + & WTOBS(K)*SORT(ABS(PRAD))
      END IF
      BDFLAG = '*'
      ELSE
      BDFLAG = ' '
      END IF
   IF (BEST) THEN
      SCNT(I, 2) = SCNT(I, 2) + 1
      SCNTWT(I, 2) = SCNTWT(I, 2) + & WTOBS(K)*SORT(ABS(PRAD))
      QCNT(IPWT + 1, 2) = QCNT(IPWT + 1, 2) + 1
      QCNTWT(IPWT + 1, 2) = QCNTWT(IPWT + 1, 2) + & WTOBS(K)*SORT(ABS(PRAD))
   END IF
   C WRITE OUT TO POLARITY FILE
   C
   WRITE (PUNIT, 250) STN(K), DIST(K), AZ(K), AIN(K),
   & PRMK(K), WTOBS(K)/WTMAX, BDFLAG
   250 FORMAT (IX, A4, 3F6.1, 3X, A4, F5.2, 2X, A1)
   CONTINUE
   WRITE (PUNIT, '*')
   END IF
   C
   C END OF SOLUTION LOOP
CONTINUE
IF (NSOL .GT. 1) WRITE (EUNIT, *) '***** EVENT ', EVENT(1:17),
& ' HAS MULTIPLE SOLUTIONS *****

EVENT
C END OF EVENT
GOTO 20
END

LOGICAL FUNCTION FPINP (CUNIT, DDELCC, DDELF, DELBC, DFI TC, DSTMX,
& DLAMC, DLAMF, DPHIC, DPHIF, ERATE, EUNIT, FMAGMN, IJEFF, IPRNT,
& IRES, JUNIT, KILSTA, MINOBS, MXDIP, MXQUAL, MXRAKE, MXSTAT,
& MXSTRK, NDELC, NDELF, NEV, NKIL, NLAMC, NLAMF, NPHIC, NPHIF,
& NREV, PHIC, REVSTA, WEIGHT, XLAMC)

C READ IN CONTROL PARAMETERS

INTEGER CUNIT  I (INPUT) LOGICAL UNIT № OF INPUT CONTROL FILE
REAL DDEL C I (OUTPUT) FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DDELF I (OUTPUT) FAULT DIP INCREMENT IN DEGREES FOR FINE SEARCH
REAL DELBC I (OUTPUT) INITIAL FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
REAL DFI TC I (OUTPUT) INCREMENT TO COARSE FIT FUNCTION
REAL DLAMC I (OUTPUT) FAULT RAKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DLAMF I (OUTPUT) FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL DPHIC I (OUTPUT) FAULT STRIKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DPHIF I (OUTPUT) FAULT STRIKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL ERATE(MXQUAL) I (OUTPUT) ASSUMED WEIGHTED ERROR RATES FOR DATA, READ FROM CONTROL CARD
INTEGER EUNIT I (INPUT) LOGICAL UNIT № OF OUTPUT OF ERROR MESSAGES
REAL FMAGMN I (OUTPUT) Minimum permitted magnitude
INTEGER IJEFF I (OUTPUT) Flag: 1(#)=DO (NOT) USE DATA WEIGHTED OUT BY JEFFREY'S WEIGHTING
INTEGER IPRNT I (OUTPUT) Flag: 1(#)=DO (NOT) PRINT OUT FIT PARAMETERS
INTEGER IRES I (OUTPUT) Flag: 1(#)=DO (NOT) RESTRICTED SEARCH
CHARACTER*1 KILSTA(MXSTAT) I (OUTPUT) IGNORED STATION NAMES
INTEGER MINOBS I (OUTPUT) Minimum number of observations required
INTEGER MXDIP I (INPUT) Maximum # of dip increments permitted
INTEGER MXQUAL I (INPUT) Maximum # of qualities permitted
INTEGER MXRAKE I (INPUT) Maximum # of rake increments permitted
INTEGER MXSTAT I (INPUT) Maximum # of stations permitted
INTEGER MXSTRK I (INPUT) Maximum # of strike increments permitted
INTEGER NDELC I (OUTPUT) Number of fault dip increments for coarse search
INTEGER NDEL F I (OUTPUT) Number of fault dip increments for fine search
INTEGER NEV I (OUTPUT) Number of events to process
INTEGER NKIL I (OUTPUT) Number of ignored stations
INTEGER NLAMC I (OUTPUT) Number of fault rake increments for coarse search
INTEGER NLAMF I (OUTPUT) Number of fault rake increments for fine search
INTEGER NPHIC I (OUTPUT) Number of fault strike increments for coarse search
INTEGER NPHIF I (OUTPUT) Number of fault strike increments for fine search
INTEGER NREV I (OUTPUT) Number of reversed stations
REAL PHIC I (OUTPUT) Initial fault strike angle in degrees for coarse search
CHARACTER*1 REVSTA(MXSTAT) I (OUTPUT) REVERSED STATION NAMES
REAL WEIGHT(MXQUAL) I (OUTPUT) WEIGHTS ASSOCIATED WITH QUALITIES
REAL XLAMC I (OUTPUT) Initial fault rake angle in degrees for coarse search

REAL DDEL CDF I DEFAULT FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
REAL DELBCDF I DEFAULT INITIAL FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
REAL DLAMFDF I DEFAULT FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL DPHICDF I DEFAULT FAULT STRIKE INCREMENT IN DEGREES FOR FINE SEARCH
CHARACTER*1 FILNAM I NAME OF CONTROL FILE
INTEGER I I DUMMY LOOP INDEX
INTEGER  ICARD
INTEGER  INDEX1
INTEGER  INDEX2
INTEGER  IOS
INTEGER  NDELCDF
INTEGER  NDELFDF
INTEGER  NLAMCDF
INTEGER  NLAMDF
INTEGER  NPHICDF
INTEGER  NPHIFDF
REAL    PHI0CDF
REAL    PHICDF
REAL    STATN
CHARACTER*4  TMP
CHARACTER*1  WT

C C SET UP DEFAULT GRID SPACING
C
PARAMETER (PHI0CDF = 0., DEL0CDF = 10., XLAM0CDF = -180.)
PARAMETER (DPHICDF = 20., DDELCDF = 20., DLAMCDF = 20.)
PARAMETER (NPHICDF = 5, NDELCDF = 5, NLAMCDF = 18)
PARAMETER (DPHIFDF = 5., DDELFDF = 5., DLAMDF = 18.)
PARAMETER (NPHIFDF = 19, NDELFDF = 19, NLAMDF = 7)

C FPINP = .TRUE.
INQUIRE (CUNIT, NAME = FILNAM)
INDEX1 = INDEX(FILNAM, ";") + 1
INDEX2 = INDEX(FILNAM, ":") + 2
WRITE (EUNIT, *) 'CONTROL FILE "', FILNAM(INDEX1:INDEX2)
INQUIRE (IUNIT, NAME = FILNAM)
INDEX1 = INDEX(FILNAM, ";") + 1
INDEX2 = INDEX(FILNAM, ":") + 2
WRITE (EUNIT, *) 'INPUT DATA FILE "', FILNAM(INDEX1:INDEX2)
ICARD = 1
READ (CUNIT, *, ERR = 1000) DISTMX, FMAGMN, MINOBS, IPRNT, IJEFF,
& NEV, DFITC
C
C CHECK PARAMETERS
C
IF (DISTMX .LE. 8.) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: DISTMX .LE. 8 ****
  FPINP = .FALSE.
ENDIF
IF (IPRNT .NE. 0 .AND. IPRNT .NE. 1) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: IPRNT .NE. 0 OR 1 ****
  FPINP = .FALSE.
ENDIF
IF (IJEFF .NE. 0 .AND. IJEFF .NE. 1) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: IJEFF .NE. 0 OR 1 ****
  FPINP = .FALSE.
ENDIF
IF (MINOBS .LT. 0) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: MINOBS LESS THAN 0 ****
  FPINP = .FALSE.
ENDIF
IF (NEV .LE. 0) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: NEV .LE. THAN 8 ****
  FPINP = .FALSE.
ENDIF
IF (DFITC .LT. 0) THEN
  WRITE (EUNIT, *) '*** FPINP ERROR: DFITC .LT. THAN 0 ****
  FPINP = .FALSE.
ENDIF
ELSE IF (DFITC .GE. 0.25) THEN

56
WRITE (EUNIT, *) '****** WARNING: DFITC TRUNCATED TO 0.25 ******'
        END IF
C
C READ IN ESTIMATED WEIGHTED ERROR RATES FOR EACH QUALITY CLASS OF HAND-
C PICKED AND MACHINE-PICKED DATA.
C
IF (.NOT. FPINP) RETURN
ICARD = ICARD + 1
READ (CUNIT, *, ERR = 1000) (ERATE(I), I = 1, MXQUAL/2)
ICARD = ICARD + 1
READ (CUNIT, *, ERR = 1000) (ERATE(I), I = MXQUAL/2 + 1, MXQUAL)
WT = 0.
C
C CONVERT ESTIMATED ERROR RATES TO WEIGHTING FACTORS
C
DO 10 I = 1, MXQUAL
   IF (ERATE(I) .LT. 0.5) THEN
      IF (ERATE(I) .LT. 0.001) ERATE(I) = 0.001
      WEIGHT(I) = 1./SQRT(ERATE(I) - ERATE(I)*ERATE(I))
   ELSE
      WEIGHT(I) = 0.5
   END IF
   WT = WT * WEIGHT(I)
10 CONTINUE
IF (WT .EQ. 0.) THEN
   WRITE (EUNIT, *) '****** FPINP ERROR: ALL WEIGHTS ARE 0 ******'
   FPINP = .FALSE.
ENDIF
C
C READ IN LIST OF REVERSED, IGNORED STATIONS
C
IF (.NOT. FPINP) RETURN
NREV = 1
NKIL = 1
ICARD = ICARD + 1
20 READ (CUNIT, 30, END = 40, ERR = 1000) STATUS, STATN
30 FORMAT (A1, IX, A4)
   IF (STATN .NE. ' ') THEN
      IF (STATUS .EQ. 'R') THEN
         REVSTA(NREV) = STATN
         NREV = NREV + 1
      ELSE IF (STATUS .EQ. 'K') THEN
         KILSTA(NKIL) = STATN
         NKIL = NKIL + 1
      ELSE
         WRITE (EUNIT, *) '****** FPINP ERROR: STATUS OF STATION ', STATN, ' .NE. "K" OR "R" ******'
         FPINP = .FALSE.
      END IF
   END IF
   ELSE IF (STATUS .EQ. 'K') THEN
      KILSTA(NKIL) = STATN
      NKIL = NKIL + 1
   ELSE IF (STATN .GT. MXSTAT) THEN
      WRITE (EUNIT, *) '****** FPINP ERROR: # OF REVERSED STATION ', MXSTAT, ' EXCEEDS ', NREV, ' ******'
      FPINP = .FALSE.
   END IF
   GOTO 20
40 NKIL = NKIL - 1
   NREV = NREV - 1
C
C READ IN OPTIONAL GRID SEARCH PARAMETERS
C
IF (.NOT. FPINP) RETURN
ICARD = ICARD + 1
READ (CUNIT, *, IOSTAT = IOS) PHI0C, PHI1, DELC, DELF, XLAM0C,
& XLAM1, DPHIC, DDEL, DLAMC, DPHIF, DDELF, DLAMF

C UNRESTRICTED SEARCH AREA
C
IF (IOS .LT. 0) THEN
IRES = 0
PHI0C = PHI0CDF
DPHIC = DPHICDF
PHI1 = PHI0C + (NPHICDF - 1)*DPHICDF
NPHIC = NPHICDF
NPHIF = NPHICDF
DELC = DELCDF
DELF = DELCDF
DEL = DELCDF + (NDELCDF - 1)*DDELCDF
NDEL = NDELCDF
NDELF = NDELCDF
XLAM0C = XLAM0CDF
DLAMC = DLAMCDF
DLAMF = DLAMCDF
XLAM1 = XLAM0CDF + (NLAMCDF - 1)*DLAMCDF
NLAMC = NLAMCDF
NLAMF = NLAMCDF

C RESTRICTED SEARCH AREA
C
ELSE IF (IOS .EQ. 0) THEN
IRES = 1
C CHECK RESTRICTED STRIKE RANGE PARAMETERS FOR CONSISTENCY
C
IF (PHI0C .GT. PHI1) THEN
TMP = PHI1
PHI1 = PHI0C
PHI0C = TMP
END IF
DPHI = PHI1 - PHI0C
IF (DPHI .LE. 0.) THEN
WRITE (EUNIT, 60) 'DPHIC' 'FPINP ERROR: DPHIC MUST BE GREATER THAN 0'
FPINP = .FALSE.
ELSE
NPHIC = INT(DPHIC/DPHIC) + 1
ENDIF

50 FORMAT (' ', '***** FPINP ERROR: ', A6, ' (- ', F6.1, '***') FPINP = .FALSE.
ENDIF
& (PHI0C .LT. PHI1) .AND.
& (PHI1 .LT. PHI0C + FLOAT(NPHICDF)*DPHICDF) THEN
WRITE (EUNIT, 50) 'PHI0C', PHI0C, PHI1, PHI0CDF, PHI1CDF +
& FLOAT(NPHICDF)*DPHICDF
ENDIF
& ) OUTSIDE RANGE OF ', F6.1, ' - ', F6.1, ' *****'
FPINP = .FALSE.
ENDIF
& (PHI1 .LT. PHI0CDF + FLOAT(NPHICDF)*DPHICDF) THEN
WRITE (EUNIT, 50) 'PHI1', PHI1, PHI0CDF, PHI1CDF +
& FLOAT(NPHICDF)*DPHICDF
FPINP = .FALSE.
ENDIF
& ) OUTSIDE RANGE OF ', F6.1, ' - ', F6.1, ' *****'
FPINP = .FALSE.
ELSE
NPHIC = INT(DPHIC/DPHIC) + 1
ENDIF

60 FORMAT (' ', '***** FPINP ERROR: ', A6, ' AS, &'* MUST BE GREATER THAN 0 *****') FPINP = .FALSE.
ELSE
NPHIC = INT(DPHIC/DPHIC) + 1
IF (NPHIC .GT. MXSTRK) THEN
  WRITE (EUNIT, 70) 'STRIKE', PHI1, PHI0C, OPHIC, MXSTRK
  FORMAT (' ', '**** FPINP ERROR: ', A6,
& ' RANGE TOO LARGE. (', F6.1, ' - ', F6.1, ')/', F6.1,
& ' + 1 GREATER THAN ', 12, ' *****)')
  FPINP = .FALSE.
END IF
END IF
IF (OPHIF .LE. 0.) THEN
  WRITE (EUNIT, 60) 'DPHIF'
  FPINP = .FALSE.
END IF

C CHECK RESTRICTED DIP RANGE PARAMETERS FOR CONSISTENCY
C
IF (DEL0C .GT. DEL1) THEN
  TMP = DEL1
  DEL1 = DEL0C
  DEL0C = TMP
END IF
DDEL = DEL1 - DEL0C
IF (DEL0C .LT. 0.) OR. (DEL0C .GT. 90.) THEN
  WRITE (EUNIT, 50) 'DEL0C', DEL0C, 0., 90.
  FPINP = .FALSE.
END IF
IF (DEL1 .LT. 0.) OR. (DEL1 .GT. 90.) THEN
  WRITE (EUNIT, 50) 'DEL1', DEL1, 0., 90.
  FPINP = .FALSE.
END IF
IF (DDELC .LE. 0.) THEN
  WRITE (EUNIT, 60) 'DDELC'
  FPINP = .FALSE.
ELSE
  NDELC = INT(DDEL/DDELC) + 1
  IF (NDELC .GT. MXSTRK) THEN
    WRITE (EUNIT, 70) 'DIP', DEL1, DEL0C, DDEL, MXSTRK
    FPINP = .FALSE.
  END IF
END IF
END IF
IF (DDELF .LE. 0.) THEN
  WRITE (EUNIT, 60) 'DDELF'
  FPINP = .FALSE.
END IF

C CHECK RESTRICTED RAKE RANGE PARAMETERS FOR CONSISTENCY
C
IF (XLAM0C .GT. XLAM1) THEN
  TMP = XLAM1
  XLAM1 = XLAM0C
  XLAM0C = TMP
END IF
DLAM = XLAM1 - XLAM0C
IF (XLAM0C .LT. 0.) OR. (XLAM0C .GT. XLAM0CDF) THEN
  WRITE (EUNIT, 50) 'XLAM0C', XLAM0C, XLAM0CDF, XLAM0CDF +
  FLOAT(NLAMCDF)*DLAMCDF
  FPINP = .FALSE.
END IF
IF (XLAM1 .LT. 0.) OR. (XLAM1 .GT. XLAM1CDF) THEN
  WRITE (EUNIT, 50) 'XLAM1', XLAM1, XLAM1CDF, XLAM1CDF +
  FLOAT(NLAMCDF)*DLAMCDF
  FPINP = .FALSE.
END IF
IF (DLAMC .LE. 0.) THEN
  WRITE (EUNIT, 60) 'DLAMC'
  FPINP = .FALSE.
ELSE
  NLAMC = INT(DLAM/OLAMC) + 1
  IF (NLAMC .GT. MXSTRK) THEN
    WRITE (EUNIT, 70) 'RAKE', XLAMI, XLAM0C, DLAMC, MXSTRK
    FPINP = .FALSE.
  END IF
ENDIF
IF (DIAMF .LE. 0.) THEN
  WRITE (EUNIT, 60) 'DLAMF'
  FPINP = .FALSE.
ENDIF

C
C READ ERROR
C ELSE
GOTO 1000
ENDIF

C
C WRITE OUT PARAMETERS
C IF (FPINP) THEN
  WRITE (EUNIT, *) 'MAXIMUM EPICENTRAL DISTANCE = ', DISTMX
  WRITE (EUNIT, *) 'MINIMUM MAGNITUDE = ', FMAGMN
  WRITE (EUNIT, *) 'MINIMUM # OBSERVATIONS = ', MINOBS
  IF (IPRNT .EQ. 1) THEN
    WRITE (EUNIT, *) 'PARAMETER FIT FILE GENERATED (IPRNT = 1)'
  ELSE
    WRITE (EUNIT, *) 'PARAMETER FIT FILE SUPPRESSED (IPRNT = 0)'
  END IF
  IF (JEFF .EQ. 1) THEN
    WRITE (EUNIT, *) 'PHASE DATA REJECTED BY JEFFREYS" WEIGHTING '
    &INCLUDED (INEF = 1)'
  ELSE
    WRITE (EUNIT, *) 'PHASE DATA REJECTED BY JEFFREYS" WEIGHTING '
    &EXCLUDED (INEF = 0)'
  END IF
  WRITE (EUNIT, *) 'UP TO ', NEV, ' EVENTS PROCESSED'
  WRITE (EUNIT, 80)
  80 FORMAT ('0', 'HAND-PICKED DATA QUALITY WEIGHT WEIGHT ERROR RATE (EST.)')
  DO 99 I = 1, MXQUAL/2
    WRITE (EUNIT, 90) I-1, WEIGHT(I), ERATE(I)
  99 CONTINUE
  WRITE (EUNIT, 110)
  110 FORMAT ('0', 'MACHINE-PICKED DATA QUALITY WEIGHT WEIGHT ERROR RATE (EST.)')
  DO 129 I = MXQUAL/2+1, MXQUAL
    WRITE (EUNIT, 90) I-MXQUAL/2, WEIGHT(I), ERATE(I)
  129 CONTINUE
ENDIF
IF (NKIL .GT. 0) THEN
  WRITE (EUNIT, 140)
  140 FORMAT ('0', 'THE FOLLOWING STATIONS IGNORED IN FAULT-PLAN '
    &CALCULATIONS'
  DO 139 I = 1, NKIL
    WRITE (EUNIT, 150)
  139 CONTINUE
ENDIF
IF (IRES .EQ. 1) THEN
  WRITE (EUNIT, 140)
ENDIF
SUBROUTINE READEQ (AIN, AZ, DIST, DISTMX, EUNIT, EVENT, FMAGMN, & IJEFF, IUNIT, KILSTA, MINOBS, MXQUAL, MXSTAT, NREV, & POBS, PRCNTX, PRMK, REVSTA, SIGMAF, STN, SUMWT, WEIGHT, WTOBS)

C READS HYP071 OUTPUT LISTING. RETURNS SUMMARY CARD AND CORRESPONDING PHASE FIRST MOTIONS, QUALITIES, ANGLES OF INCIDENCE, STATION NAMES, AND AZIMUTHS. CALCULATES STANDARD DEVIATION (SIGMAF) OF FIT FROM ESTIMATED STANDARD DEVIATIONS OF THE DATA. THE ESTIMATED DATA ERRORS ARE CONTROL-FILE INPUTS; CORRESPONDING DATA WEIGHTS ARE CALCULATED IN MAIN AND PASSED TO THIS ROUTINE IN THE PARAMETER "WEIGHT".

C

REAL AIN(MXSTAT)
REAL AZ(MXSTAT)
REAL DIST(MXSTAT)
REAL DISTMX
INTEGER EUNIT
CHARACTER*80 EVENT
REAL FMAGMN
INTEGER IJEFF
INTEGER IUNIT
CHARACTER*4 KILSTA(MXSTAT)
INTEGER MINOBS
INTEGER MXQUAL
INTEGER MXSTAT
INTEGER NKIL
INTEGER NR
INTEGER NREV
REAL POBS(MXSTAT)
REAL PRCNTX
CHARACTER*4 PRMK(MXSTAT)
CHARACTER*4 REVSTA(MXSTAT)
REAL SIGMAF
REAL STN(MXSTAT)
REAL SUMWT
REAL WEIGHT(MXQUAL)
REAL WTOBS(MXSTAT)

C CHARACTER*1 FM
REAL FMAG
INTEGER IPWT
INTEGER J
CHARACTER*2 JFRYWT
INTEGER JWT
INTEGER K
CHARACTER*8# LINE
CHARACTER*1 M
INTEGER NCLAS(2#)
CHARACTER*4 TEST
REAL VARF
REAL WT

C RESET VALUES
C
DO 10 I = 1, MXQUAL
NCLAS(I) = 0
10 CONTINUE
C
C FIND LINE PRIOR TO SUMMARY CARD
C
PRCNTX = 0.
SUMWT = 0.
20 READ (IUNIT, 30, END = 1000) EVENT
FORMAT (2X, A4)
READ (EVENT, *(45X, F5.2)) FMAG
C
C CHECK MAGNITUDE
C
IF (FMAG .LT. FMAGMN) THEN
NR = 0
RETURN
END IF
READ (IUNIT, 30, END = 1000) EVENT
FORMAT (IX, A80)
READ (EVENT, *(45X, F5.2)) FMAG
C
C CHECK MAGNITUDE
C
IF (FMAG .LT. FMAGMN) THEN
NR = 0
RETURN
END IF
READ (IUNIT, 30, END = 1000) EVENT
READ (IUNIT, 30, END = 1000) EVENT
C
C CHECK WHETHER PHASE DATA OR FOCAL MECHANISM FOLLOW
C
IF (TEST .NE. 'STN ') GOTO 20
K = I
50 STN(K) = ' '
READ (IUNIT, 60, END = 70) LINE

60 FORMAT (A)
READ (LINE, "(A1, A4)") M, STN(K)

C CHECK FOR END OF PHASE DATA
C
70 IF (M .EQ. '!' .OR. STN(K) .EQ. ' ') THEN
C END OF EVENT
C
IF (K .GE. MINOBS) THEN
NR = K - 1
PRCNTX = PRCTNX / FLOAT(NR)
VARF1 = 0.
VARF2 = 0.
DO 80JWT = 1, MXQUAL
   IF (WEIGHT(JWT) .GE. .0001) THEN
      VARF1 = VARF1 + NCLAS(JWT)
      VARF2 = VARF2 + NCLAS(JWT) * WEIGHT(JWT)
   END IF
80 CONTINUE
VR = VARFI / (VARF2 + 0.01)
SIGMAF = SQRT(VARF)
ELSE
NR = 0
END IF
RETURN
END IF

C CHECK FOR REPEATED PHASE CARD
C
IF (K .GT. 2) THEN
   DO 100 J = 1, K - 1
      IF (STN(K) .EQ. STN(J)) GOTO 90
100 CONTINUE
END IF

C IGNORE THIS STATION?
C
IF (NKIL .GT. 0) THEN
   DO 110 I = 1, NKIL
      IF (STN(K) .EQ. KILSTA(I)) GOTO 50
110 CONTINUE
END IF

C SO FAR, SO GOOD: NOW CHECK PHASE CARD FOR POLARITY, DISTANCE, JEFFREYS' WEIGHTING, QUALITY
C
READ (LINE, 120) DIST(K), AZ(K), AIN(K), PRMK(K), JFRYWT
120 FORMAT (F5.1, 1X, F3.0, 1X, F3.8, 1X, A4, 35X, A2)
READ (PRMK(K), ' (2X, A1, I1)') FM, IPWT
IF (FM .NE. 'U' .AND. FM .NE. 'D' .AND. FM .NE. '+' .AND.
   & FM .NE. '-') GOTO 50
IF (DIST(K) .GT. DISTMX) GOTO 50
IF (JEFF .EQ. 0 .AND. JFRYWT .EQ. ' ') GOTO 50
IF (IPWT .GE. MXQUAL) THEN
   WT = 0.
ELSE IF (PRMK(K)(1:1) .EQ. 'X' .OR. PRMK(K)(1:1) .EQ. 'Y') THEN
   JWT = JWT + MXQUAL / 2 + 1
   WT = WEIGHT(JWT)
ELSE
   IF (IPWT .GE. MXQUAL) THEN
      WT = 0.
   ELSE
      WT = WT + 1
   END IF
IF (JWT .GE. MXQUAL) THEN
   JWT = JWT + MXQUAL / 2 + 1
   WT = WEIGHT(JWT)
ENDIF
IF (WT .NE. 0.) PRCNTX = PRCNTX + 1.
ELSE
JWT = IPWT + 1
WT = WEIGHT(JWT)
END IF
IF (WT .EQ. 0.) GOTO 58
C
C FLIP POLARITIES IF STATION IS DESIGNATED AS REVERSED
C
DO 130 I = 1, NREV
IF (STN(K).EQ. REVSTA(I)) THEN
  IF (FM .EQ. 'U') PRMK(K)(3) = -D'
  IF (FM .EQ. 'D') PRMK(K)(3) = '+
  IF (FM .EQ. '*') PRMK(K)(3) = '-'
  IF (FM .EQ. '-') PRMK(K)(3) = '+'
  FM = PRMK(K)(3)
END IF
130 CONTINUE
NCLAS(JWT) = NCLAS(JWT) + 1
WTOS(K) = WT
SUMWT = SUMWT + WT
IF (FM .EQ. 'U' .OR. FM .EQ. '+') THEN
  POBS(K) = .5
ELSE
  POBS(K) = -.5
END IF
C
C INCREMENT K AND CHECK NUMBER AGAINST ARRAY DIMENSIONS
C
K = K + 1
IF (K.GT. MXSTAT) THEN
  WRITE (EUNIT, '*') '*** READ ERROR: NUMBER OF STATIONS READING EXCEEDS ', MXSTAT, ' FOR EVENT ', EVENT(I), '*
&NGS EXCEEDS ', MXSTAT, ' FOR EVENT ', EVENT(I), '***
NR = K - 1
PRCNTX = PRCNTX/FLOAT(NR)
ELSE
NR = 0
END IF
RETURN
END IF
C
C READ ANOTHER PHASE
C
GOTO 58
C
C END OF FILE
C
1000 NR = -1
RETURN
END
C
C SUBROUTINE FOUT(DDELF, DLAMF, DPHIF, ERATE, EUNIT, IEVP, IERV, & IND, IRES, MXDIP, MXQUAL, MXRAKE, MXSTAT, MXSTRK, MXDIP, MXRNG, & NFT, NLAMF, NPHIF, NREV, NRRNG, NSRNG, NSTAT, QCNT, QCNTWT, & REVSTA, SCNT, SCNTWT, STAT)
C
C GENERATE SUMMARY LISTING OF POLARITY DISCREPANCIES AS A FUNCTION OF STATION AND QUALITY, THE DISTRIBUTION OF C FIT PARAMETERS, AND THE DISTRIBUTION OF DIP, STRIKE, AND RAKE RANGES ABOUT BEST FIT SOLUTION
REAL DDELF
REAL DLAMF
REAL DPHIF
REAL RATE(MXQUAL)
INTEGER EUNIT
INTEGER IEVP
INTEGER IEVR
INTEGER IND(MXSTAT)
INTEGER IRES
INTEGER MXDIP
INTEGER MXQUAL
INTEGER MXRAKE
INTEGER MXSTAT
INTEGER MXSTRK
INTEGER NDELF
INTEGER NDRNG(MXDIP)
INTEGER NFIT(2)
INTEGER NLAFT
INTEGER NPHIF
INTEGER NREV
INTEGER NRRNG(MXRAKE)
INTEGER NSTAT
INTEGER QCNT(MXQUAL, 2)
REAL QCNTU(MXQUAL, 2)
CHARACTER*4 REVSTA(MXSTAT)
CHARACTER*4 STAT(MXSTAT)
CHARACTER*2 ESTAR
INTEGER I
INTEGER J
INTEGER NATOT
INTEGER NTOT
REAL RATE
CHARACTER*1 STAR
REAL WTOT

C REAL DDELF
C REAL DLAMF
C REAL DPHIF
C REAL RATE(MXQUAL)
C INTEGER EUNIT
C INTEGER IEVP
C INTEGER IEVR
C INTEGER IND(MXSTAT)
C INTEGER IRES
C INTEGER MXDIP
C INTEGER MXQUAL
C INTEGER MXRAKE
C INTEGER MXSTAT
C INTEGER MXSTRK
C INTEGER NDELF
C INTEGER NDRNG(MXDIP)
C INTEGER NFIT(2)
C INTEGER NLAFT
C INTEGER NPHIF
C INTEGER NREV
C INTEGER NRRNG(MXRAKE)
C INTEGER NSTAT
C INTEGER QCNT(MXQUAL, 2)
C REAL QCNTU(MXQUAL, 2)
C CHARACTER*4 REVSTA(MXSTAT)
C CHARACTER*4 STAT(MXSTAT)
C CHARACTER*2 ESTAR
C INTEGER I
C INTEGER J
C INTEGER NATOT
C INTEGER NTOT
C REAL RATE
C CHARACTER*1 STAR
C REAL WTOT

C NDTOT = #
C NTOT = #
C WTOT = #

DO 5 I = 1, MSTAT
   NDTOT = NDTOT + QCNT(I, 1)
   NTOT = NTOT + QCNT(I, 2)
   WTOT = WTOT + QCNTU(I, 2)
5 CONTINUE

NATOT = NTOT - NDTOT

C WRITE OUT SUMMARY OF POLARITY DISCREPANCIES BY STATION
C WRITE (EUNIT, 16) 
C 16 FORMAT ('* ', 'SUMMARY OF STATIONS HAVING POLARITIES IN DISCREPANCY & WITH BEST FIT SOLUTION (* DENOTES REVERSED STATION)', /
C & ' STATION DISCREPANCIES AGREEMENTS TOTAL WEIGHTED ERROR RATE TOTAL ERROR CONTRIBUTION', /
C C C SORT STATIONS ALPHABETICALLY
C CALL CSORT (STAT, IND, MSTAT)
DO 46 I = 1, MSTAT
   J = IND(I)
STAR = ' '
DO 20 K = 1, NREV
   IF (STAT(J) .EQ. REVSTA(K)) STAR = '*'
20 CONTINUE
WRITE (EUNIT, 30) STAR, STAT(J), SCNT(J, 1), SCNT(J, 2) -
   & SCNT(J, 1), SCNT(J, 2), SCNTWT(J, 1)/SCNTWT(J, 2),
   & SCNTWT(J, 1)/WTOT
30 FORMAT (' ', A1, A4, 3(I4, X), 9X, F6.3, 10X, F6.4)
4# CONTINUE
WRITE (EUNIT, 50) NDTOT, NATOT, NTOT
50 FORMAT (' ', 'TOTAL ', 3(I4, X), 15)
C WRITE OUT SUMMARY OF HAND-Picked POLARITY DISCREPANCIES BY READING QUALITY
C WRITE (EUNIT, 70)
70 FORMAT (' ', 'SUMMARY OF HAND-Picked DATA WITH RESPECT TO
   & BEST FIT SOLUTIONS', '/',
   & ' QUAL DISCREPANCIES AGREEMENTS TOTAL
   & WEIGHTED ERROR RATE', '/)
NDTOT = 0
NTOT = 0
DO 90 I = 1, MXQUAL/2
   ESTAR = '*'
   NDTOT = NDTOT + QCNT(I, 1)
   NTOT = NTOT + QCNT(I, 2)
   IF (QCNTWT(I, 2) .EQ. 0.) THEN
      RATE = 0.
   ELSE
      RATE = QCNTWT(I, 1)/QCNTWT(I, 2)
      IF (RATE .GE. 0.0001) THEN
         IF (ABS((ERATE(I) - RATE)/RATE) .GE. 0.2) ESTAR = '**'
      END IF
   END IF
   WRITE (EUNIT, 80) I - 1, QCNT(I, 1), QCNT(I, 2) -
   & QCNT(I, 1), QCNT(I, 2), RATE, ESTAR
90 CONTINUE
NATOT = NTOT - NDTOT
WRITE (EUNIT, 50) NDTOT, NATOT, NTOT
C WRITE OUT SUMMARY OF MACHINE-PICKED POLARITY DISCREPANCIES BY READING QUALITY
C WRITE (EUNIT, 110)
110 FORMAT (' ', 'SUMMARY OF MACHINE-PICKED DATA WITH RESPECT TO
   & BEST FIT SOLUTIONS', '/',
   & ' QUAL DISCREPANCIES AGREEMENTS TOTAL
   & WEIGHTED ERROR RATE', '/)
NDTOT = 0
NTOT = 0
DO 120 I = MXQUAL/2 + 1, MXQUAL
   ESTAR = '*'
   NDTOT = NDTOT + QCNT(I, 1)
   NTOT = NTOT + QCNT(I, 2)
   IF (QCNTWT(I, 2) .EQ. 0.) THEN
      RATE = 0.
   ELSE
      RATE = QCNTWT(I, 1)/QCNTWT(I, 2)
      IF (RATE .GE. 0.0001) THEN
         IF (ABS((ERATE(I) - RATE)/RATE) .GE. 0.2) ESTAR = '**'
      END IF
   END IF
   WRITE (EUNIT, 80) I - MXQUAL/2 - 1, QCNT(I, 1), QCNT(I, 2) -
   & QCNT(I, 1), QCNT(I, 2), RATE, ESTAR
120 CONTINUE
NATOT = NTOT - NDTOT
WRITE (EUNIT, 50) NDTOT, NATOT, NTOT
C WRITE OUT DISTRIBUTION OF FIT PARAMETERS
C WRITE (EUNIT, 130)
130 FORMAT ('#', 'DISTRIBUTION OF SOLUTION MISFIT SCORES', /,
  & IX, ' MISFIT SCORE *')
DO 150 I = 1, 20
  WRITE (EUNIT, 140) FLOAT(I)*.025, NFIT(I)
140 FORMAT (*', F5.3, * ', F5.3, 3X, 15)
150 CONTINUE
C WRITE OUT DISTRIBUTION OF SOLUTION MISFIT SCORES
C WRITE (EUNIT, 130)
130 FORMAT ('#', 'DISTRIBUTION OF SOLUTION DIP RANGES', /,
  & IX, ' RANGE *')
DO 180 I = 1, NDELF
  WRITE (EUNIT, 170) FLOATU - 1>*DDELF, NDRNG(I)
170 FORMAT (' ', IX, F5.1, 5X, 15)
180 CONTINUE
WRITE (EUNIT, 190)
190 FORMAT ('0', 'DISTRIBUTION OF SOLUTION STRIKE RANGES', /,
  & IX, ' RANGE *')
DO 220 I = 1, NPHIF
  WRITE (EUNIT, 170) FLOAT(I - 1)*DPHIF, NSRNG(I)
220 CONTINUE
WRITE (EUNIT, 210)
210 FORMAT ('0', 'DISTRIBUTION OF SOLUTION RAKE RANGES', /,
  & IX, ' RANGE *')
DO 250 I = 1, NRRNG
  WRITE (EUNIT, 170) FLOAT(I)*DLAMF, NRRNG(I)
250 CONTINUE
END IF
C WRITE (EUNIT, *) ' '
WRITE (EUNIT, *) IEVR, ' EVENTS READ, ', IEVP, ' PROCESSED'
C RETURN
END
C SUBROUTINE CSORT(CX, IX, N)
C INDIRECT SORT ROUTINE FROM MEISSNER & ORGANICK, P.352
C STORES ASCENDING ORDER OF CX IN ARRAY IX, LEAVING CX UNCHANGED
C INTEGER IX(*)
C CHARACTER*(*) CX(*)
C INTEGER N
C INTEGER I
C INTEGER J
C INTEGER NEXT
C DO 19 I = 1, N
  19 IX(I) = I
C DO 49 J = 1, N - 1
SUBROUTINE SEARCH (BOT, COEF, DDEL, DEL, DELC, DEL#, DFIT, DLAM,
& DPHI, FIRST, FIT, FITMIN, FLAG, GFIT, IGOOD, IPRMT, J1, M1,
& MXDIP, MXRAKE, MXSTAT, MXSTRK, M1, NDEL, NG, NLAM, NPHI, NR,
& NSTAR, PHIS, PHISC, PHIS#, PPOS, RAD, WTOBS, XLAM, XLAMC, XLAM#)

C LOOP OVER THE ENTIRE FOCAL MECHANISM SPACE, COMPUTE FIT PARAMETER FOR EACH SOLUTION, AND RETURN BEST FIT INDICES
C IN CASE OF TIE FIT, CHOOSE FIT WITH LARGEST "BOT". IF FINE SEARCH (FIRST = FALSE) THEN FILL IN FLAG ARRAY WITH
C STARS FOR SOLUTIONS WITH FIT PARAMETER <= BEST FIT + DFIT
C
REAL BOT(MXDIP,MXSTRK,MXRAKE) I (OUTPUT) SUM OF PRODUCT OF OBSERVED AND PREDICTED WEIGHTS
REAL COEF(MXSTAT,6) I (INPUT) COEFFICIENTS BY WHICH TM MULTIPLIED TO GIVE P RADIATION PATTERN
REAL DDEL(MXDIP) I (OUTPUT) FAULT DIP ANGLE IN DEGREES
REAL DELC(MXDIP) I (OUTPUT) FAULT DIP ANGLE FOR COARSE SEARCH
REAL DEL# I (INPUT) INITIAL FAULT DIP ANGLE IN DEGREES
REAL DFIT I (INPUT) INCREMENT TO FIT FUNCTION
REAL DLAB I (INPUT) FAULT RAKE INCREMENT IN DEGREES
REAL DPHI I (INPUT) FAULT STRIKE INCREMENT IN DEGREES
LOGICAL FIRST I (INPUT) FIRST TIME INTO SUBROUTINE SEARCH
REAL FIT(MXDIP,MXSTRK,MXRAKE) I (OUTPUT) WEIGHTED MEASURE OF AGREEMENT BETWEEN OBS, PRED POLARITIES
REAL FITMIN I (OUTPUT) FIT OF BEST SOLUTION CORRESPONDING TO FIT(J1, M1, M1)
REAL GFIT(MXDIP,MXSTRK,MXRAKE) I (OUTPUT) FITS OF GOOD SOLUTIONS FROM COARSE SEARCH
INTEGER IGOOD(MXDIP,MXSTRK,MXRAKE,4) I (OUTPUT) ARRAY CONTAINING INDICES OF GOOD SOLUTIONS (COARSE)
INTEGER IPRMT I (OUTPUT) FLAG: I(1)=DO (NOT) PRINT OUT FIT PARAMETERS
INTEGER J1 I (OUTPUT) DIP INDEX OF BEST SOLUTION
INTEGER M1 I (OUTPUT) RAKE INDEX OF BEST SOLUTION
INTEGER MXDIP I (INPUT) MAXIMUM # OF DIP INCREMENTS PERMITTED
INTEGER MXRAKE I (INPUT) MAXIMUM # OF RAKE INCREMENTS PERMITTED
INTEGER MXSTAT I (INPUT) MAXIMUM # OF STATIONS PERMITTED
INTEGER MXSTRK I (INPUT) MAXIMUM # OF STRIKE INCREMENTS PERMITTED
INTEGER NI I (OUTPUT) STRIKE INDEX OF BEST SOLUTION
INTEGER NDEL I (INPUT) NUMBER OF FAULT DIP INCREMENTS
INTEGER NLAM I (INPUT) NUMBER OF FAULT RAKE INCREMENTS
INTEGER NPHI I (INPUT) NUMBER OF FAULT STRIKE INCREMENTS
INTEGER NR I (INPUT) -I=EOF, #={NR<MINOBS, NR}> NUMBER OF STATIONS
INTEGER NSTAR I (OUTPUT) NUMBER OF SOLUTIONS HAVING FIT WITHIN 5X OF FITMIN
REAL PHIS(MXSTRK) I (OUTPUT) FAULT STRIKE ANGLE IN DEGREES
REAL PHISC(MXSTRK) I (OUTPUT) FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
REAL PHIS# I (INPUT) INITIAL FAULT STRIKE ANGLE IN DEGREES
REAL PPOS(MXSTAT) I (INPUT) OBSERVED FIRST MOTION POLARITIES; .5=COMPRESSION, -.5=DILATATION
REAL RAD I (INPUT) CONVERSION FROM DEGREES TO RADIANS
REAL WTOBS(MXSTAT) I (INPUT) OBSERVED FIRST MOTION WEIGHTS
REAL XLAM(MXRAKE) I (OUTPUT) FAULT RAKE ANGLE IN DEGREES
REAL XLAMC(MXRAKE) I (OUTPUT) FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
REAL XLAM# I (INPUT) INITIAL FAULT RAKE ANGLE IN DEGREES
C REAL BEST I LARGEST BOT FOR SOLUTIONS WITH FIT+FITMIN (IE. TIES)
REAL DIP
LOGICAL FIRST1
REAL FITLIM
INTEGER J
INTEGER M
INTEGER N
REAL PRAD
REAL PTH
REAL SLIP
REAL STRIKE
REAL TM(6)
REAL TOP
REAL WTTH

1 FAULT DIP ANGLE IN RADIANS
1 TEST FOR FIRST TIME INTO ROUTINE
1 UPPER BOUND ON “GOOD” SOLUTIONS IN SEARCH
1 LOOP INDEX OVER DIP
1 LOOP INDEX OVER RAKE
1 LOOP INDEX OF STRIKE
1 RADIATION AMPLITUDE CORRESPONDING AIN, PHI.
1 (DILATATION) = -1.(PRAD(+1)(COMPRESSION)
1 PREDICTED FIRST MOTION POLARITY; SAME CONVENTION AS FOR POBS
1 FAULT SLIP ANGLE IN RADIANS
1 FAULT STRIKE ANGLE IN RADIANS
1 MOMENT TENSOR IN UPPER TRIANGULAR SYMMETRIC STORAGE MODE
1 SUM OF WEIGHTED DIFFERENCE OF PREDICTED, OBS. POLARITIES; 0 <= TOP <= 1
1 PREDICTED FIRST MOTIONS WEIGHTS

C BEST = 0.
FITMIN = 2.0
DO 50 M = 1, NLAM
XLAM(M) = XLAM0 + (M - 1)*DLAM(1)
IF (FIRST) XLAMC(M) = XLAM(M)

DO 40 N = 1, NPHI
PHIS(N) = PHIS0 + (N - 1) * DPMI(1)
IF (FIRST) PHISC(N) = PHIS(N)

DO 30 J = 1, NDEL
DEL(J) = DEL0 + (J - 1) * DDEL(1)
IF (FIRST) DELC(J) = DEL(J)

STRIKE = PHIS(N)*RAD
DIP = DEL(J)*RAD
SLIP = XLAM(M)*RAD

C CALCULATE MOMENT TENSOR REPRESENTATION OF SHEAR FAULT (POSITIVE UP, SOUTH, EAST)
C CALL SHRFLT (STRIKE, DIP, SLIP, TM)
C CALCULATE RADIATION PATTERN FOR MODEL (EQN 4.91, AKI & RICHARDS, PG. 118)
C
TOP = 0
BOT(J, N, M) = 0
DO 20 I = 1, NR
PRAD = 0
DO 10 K = 1, 6
PRAD = PRAD + TM(K)*COEF(I, K)
10 CONTINUE
20 CONTINUE

FIT(J, N, M) = SIGN(0.5, PRAD)
WTTH = SORT(ABS(PRAD))
TOP = TOP + ABS(POBS(I) - PTH)*WTTH(I)*WTTH
BOT(J, N, M) = BOT(J, N, M) + WTTH(I)*WTTH

FIT(J, N, M) = TOP/BOT(J, N, M)

IF (FIT(J, N, M).LT. FITMIN) FITMIN = FIT(J, N, M)

FOR THE SOLUTIONS, FIND SOLUTION WITH MOST STATIONS AWAY FROM NODES
FITLIM = FITMIN + DFIT
NG = 0
NSTAR = -1
DO 90 M = 1, NLAM
DO 80 N = 1, NPHI
DO 70 J = 1, NDEL
  IF (FIT(J, N, M) .EQ. FITMIN .AND. BOT(J, N, M) .GT. BEST)
    & THEN
    BEST = BOT(J, N, M)
    J1 = J
    M1 = N
    M1 = M
    END IF
C
C SAVE SOLUTIONS IN COARSE SEARCH WITH FIT .LE. FITLIM AS "GOOD" SOLUTIONS
C
  IF (FIRST) THEN
    IF (FIT(J, N, M) .LE. FITLIM) THEN
      NG = NG + 1
      IGOOD(NG, 1) = J
      IGOOD(NG, 2) = N
      IGOOD(NG, 3) = M
      IGOOD(NG, 4) = 0
      GFIT(NG) = FIT(J, N, M)
    END IF
  C
  C STAR SOLUTIONS HAVING FIT WITHIN DFIT OF FITMIN ON FINE SEARCH
  C
  ELSE
    IF (FIT(J, N, M) .LE. FITLIM) THEN
      FLAG(J, N, M) = '*'
      NSTAR = NSTAR + 1
    ELSE
      FLAG(J, N, M) = ' ' 
    END IF
  END IF
70 CONTINUE
80 CONTINUE
90 CONTINUE

IF (.NOT. FIRST) FLAG(J1, N1, M1) = 'A'
C
RETURN
C
END
C
C
C SUBROUTINE PEXCF (COEF, I, MXSTAT, U)
C
C CALCULATES COEFFICIENTS FOR DETERMINING THE FAR-FIELD RADIATION PATTERN OF P WAVES FROM THE MOMENT-RATE TENSOR COMPONENTS OF A
C POINT SOURCE IN AN INFINITE, HOMOGENEOUS, ELASTIC MEDIUM. THE RADIATION PATTERN IS NORMALIZED; TO OBTAIN PARTICLE AMPLITUDES,
C MULTIPLY BY
C
1.0/(4.0*PI*RH0*(V**3)*R),
C
WHERE:
C
RH0 IS THE DENSITY IN THE SOURCE REGION,
V IS THE P-WAVE SPEED IN THE SOURCE REGION, AND
R IS THE GEOMETRIC SPREADING FACTOR
(FOA A HOMOGENEOUS MEDIUM, THIS IS EQUAL TO THE DISTANCE
TO THE OBSERVATION POINT.)
C
REFERENCE:
C
AKI, KEIITI, AND PAUL G. RICHARDS, QUANTITATIVE SEISMOLOGY,
C FREEMAN, SAN FRANCISCO, 1980, EQUATION 49.1, PAGE 118.
C
C WRITTEN BY BRUCE JULIAN
REAL COEF(MXSTAT, 6)       1 (OUTPUT) EXCITATION COEFFICIENTS
INTEGER I             1 (INPUT) INDEX OF STATION
INTEGER MXSTAT        1 (INPUT) MAXIMUM # OF STATIONS PERMITTED
REAL U(3)               1 (INPUT) UNIT VECTOR IN RAY DIRECTION

COEF(I, 1) = U(1)*U(1)
COEF(I, 2) = 2.*U(1)*U(2)
COEF(I, 3) = U(2)*U(2)
COEF(I, 4) = 2.*U(3)*U(1)
COEF(I, 5) = 2.*U(2)*U(3)
COEF(I, 6) = U(3)*U(3)

RETURN
END

SUBROUTINE SHRFLT(STRIKE, DIP, SLIP, TM)

METHOD
THE MOMENT TENSOR IS FIRST EXPRESSED IN A COORDINATE SYSTEM WITH THE Z AXIS NORMAL TO THE FAULT PLANE AND THE X AXIS IN THE SLIP DIRECTION:

\[
\begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
\end{pmatrix}
\]

THIS COORDINATE SYSTEM IS THEN ROTATED THROUGH THE EULER ANGLES PHI = -SLIP, THETA = -DIP, AND PSI = STRIKE - PI, (CONVENTIONS OF GOLDSTEIN, CLASSICAL MECHANICS, SEC 4-4) WHICH RESULTS IN A (SOUTH, EAST, UP) ORIENTATION OF THE (X, Y, Z) AXES, RESPECTIVELY. A PERMUTATION THEN CONVERTS THIS TO THE ORDER (UP, SOUTH, EAST). THE STRENGTH OF THE DOUBLE-COUPLE IS TAKEN AS UNITY; THE CALCULATED MOMENT TENSOR COMPONENTS MUST BE MULTIPLIED BY THE FACTOR:

\[\mu*A*S\]

WHERE:
\(\mu\) IS THE RIGIDITY MODULUS OF THE MEDIUM
A IS THE FAULT AREA, AND
S IS THE MEAN DISLOCATION ACROSS THE FAULT.

NOTE: IF THE MEAN DISLOCATION VELOCITY IS USED INSTEAD, THE RESULT WILL BE THE MOMENT-RATE TENSOR.)

WRITTEN BY BRUCE R. JULIAN ON 7 APRIL, 1977.

REAL DIP           1 (INPUT) FAULT DIP ANGLE IN RADIANS
REAL SLIP          1 (INPUT) FAULT SLIP ANGLE IN RADIANS
REAL STRIKE        1 (INPUT) FAULT STRIKE ANGLE IN RADIANS
REAL TM(6)         1 (OUTPUT) SEISMIC MOMENT TENSOR ARRANGED IN THE FOLLOWING ORDER:
                   
                   | (R, R) | I.E. (UP, UP) |
                   | (R, THETA) | I.E. (UP, SOUTH) |
                   | (THETA, THETA) | I.E. (SOUTH, SOUTH) |
                   | (R, PHI) | I.E. (UP, EAST) |
                   | (THETA, PHI) | I.E. (SOUTH, EAST) |
                   | (PHI, PHI) | I.E. (EAST, EAST) |
                   |

REAL A11            1 TRANSFORMATION MATRIX
REAL A21            1 TRANSFORMATION MATRIX
REAL A31            1 TRANSFORMATION MATRIX
REAL A12            1 TRANSFORMATION MATRIX
REAL A22            1 TRANSFORMATION MATRIX
REAL A32            1 TRANSFORMATION MATRIX
REAL A13            1 TRANSFORMATION MATRIX
REAL A23            1 TRANSFORMATION MATRIX
REAL A33  TRANSFORMATION MATRIX
REAL CD  |  COS(DIP)
REAL CL  |  COS(SLIP)
REAL CS  |  COS(STRIKE)
REAL SD  |  SIN(DIP)
REAL SL  |  SIN(SLIP)
REAL SS  |  SIN(STRIKE)

C CALCULATE COMPONENTS OF ORTHOGONAL TRANSFORMATION MATRIX
C FROM FAULT-ORIENTED TO (SOUTH, EAST, UP) COORDINATE SYSTEM

SS = SIN(STRIKE)
CS = COS(STRIKE)
SD = SIN(DIP)
CD = COS(DIP)
SL = SIN(SLIP)
CL = COS(SLIP)

A11 = -CS*CL - CD*SL*SS
A21 = SS*CL - CD*SL*CS
A31 = SD*SL
A13 = SS*SD
A23 = CS*SD
A33 = CD

TRANSFORM MOMENT TENSOR (0, 0, 1, 0, 0, 0)
AND PERMUTE AXES TO (UP, SOUTH, EAST) ORDER

TM(1) = 2*A31*A33
TM(2) = A11*A33 + A31*A13
TM(3) = 2*A11*A13
TM(4) = A21*A33 + A31*A23
TM(5) = A11*A23 + A21*A13
TM(6) = 2*A21*A23

RETURN
END

SUBROUTINE REFRMT (DEL, IDIP, IDIPDR, ISLIP, PHIS, XLAM)

REFORMAT DIP, STRIKE, AND RAKE ANGLES TO INTEGER VALUES AND CONVERT STRIKE TO DOWN-DIP DIRECTION

REAL DEL
INTEGER IDIP
INTEGER IDIPDR
INTEGER ISLIP
REAL PHIS
REAL XLAM

ISTRK = ISTRK + 1B0

IF (IDIP .GT. 90) THEN
IDIP = 1B0 - IDIP
ISTRK = ISTRK + 1B0
ISLIP = -ISLIP

RETURN
END
ELSE IF (IDIP < 0) THEN
   IDIP = -IDIP
   ISTRK = ISTRK + 180
   ISLIP = ISLIP + 180
END IF

IDIPDR = MOD(IDIP, 360)
IF (IDIPDR < 0) IDIPDR = IDIPDR + 360
ISLIP = MOD(ISLIP, 360)
IF (ISLIP < 180) ISLIP = ISLIP + 360
IF (ISLIP > -180) ISLIP = ISLIP - 360
RETURN
END

LOGICAL FUNCTION COMPL (SOLNS, NSOL, DD, DA, SA, AERR, MXSLNS)

THIS FUNCTION COMPARES A "NEW" FAULT PLANE SOLUTION (DD, DA, SA) WITH A LIST OF OTHER FAULT PLANE SOLUTIONS
AND CHECKS FOR ANY OF THE FOLLOWING CONDITIONS
1. THE "NEW" SOLUTION IS SIMILAR TO ONE OF THE SOLUTIONS IN SOLNS
2. THE COMPLIMENT OF THE "NEW" SOLUTION IS SIMILAR TO ONE OF THE SOLUTIONS IN SOLNS
3. THE "NEW" SOLUTION IS SIMILAR TO THE COMPLIMENT OF ONE OF THE SOLUTIONS IN SOLNS
IF ANY ONE OF THE ABOVE CONDITIONS IS TRUE, FUNCTION COMPL RETURNS WITH A VALUE .TRUE.
OTHERWISE, FUNCTION COMPL RETURNS WITH THE VALUE .FALSE.

SOLUTIONS ARE SIMILAR IF ALL THREE PAIRS OF CORRESPONDING ANGLES DIFFER BY LESS THAN AERR.

REAL AERR 1 ALLOWABLE DIFFERENCE BETWEEN CORRESPONDING ANGLES OF COMPLIMENTARY PLANES
REAL DA 1 DIP ANGLE OF NEW PLANE
REAL DD 1 DIP DIRECTION ANGLE OF NEW PLANE
INTEGER MXSLNS 1 MAXIMUM # OF MULTIPLE SOLUTIONS PERMITTED
INTEGER NSOL 1 NUMBER OF PLANES STORED IN ARRAY SOLNS
REAL SA 1 SLIP ANGLE OF NEW PLANE
REAL SOLNS(MXSLNS,3) 1 ARRAY OF PLANES (DD, DA, SA) TO TEST NEW PLANE AGAINST

INTEGER J 1 LOOP INDEX
REAL AUXI(3) 1 DIP DIRECTION, ANGLE, AND RAKE OF AUXILLIARY PLANE OF NEW PLANE
REAL AUX2(3) 1 DIP DIRECTION, ANGLE, AND RAKE OF AUXILLIARY PLANE OF PREVIOUS PLANES

COMPL = .FALSE.

CALL AUXPLN (DD, DA, SA, AUX1(1), AUX1(2), AUX1(3))

DO 40 J = 1, NSOL

COMPARE NEW SOLUTION WITH EACH SOLUTION ON LIST

IF (ABS(DD - SOLNS(J, 1)) .LT. AERR .AND.
ABS(DA - SOLNS(J, 2)) .LT. AERR .AND.
&RDIF(SA, SOLNS(J, 3)) .LT. AERR) THEN
   COMPL = .TRUE.
   RETURN
END IF

COMPARE COMPLIMENT OF "NEW SOLUTION" WITH EACH SOLUTION ON LIST

IF (ABS(SOLNS(J, 1) - AUX1(1)) .LT. AERR .AND.
ABS(SOLNS(J, 2) - AUX1(2)) .LT. AERR .AND.
&RDIF(SOLNS(J, 3), AUX1(3)) .LT. AERR) THEN

76
SUBROUTINE AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)

CALCULATE THE AUXILIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.

WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)


REAL  DA1 (INPUT) DIP ANGLE IN DEGREES OF PRINCIPLE PLANE
REAL  DD1 (INPUT) DIP DIRECTIONS IN DEGREES OF PRINCIPLE PLANE
REAL  SA1 (INPUT) SLIP ANGLE IN DEGREES OF PRINCIPLE PLANE
REAL  DA2 (OUTPUT) DIP ANGLE IN DEGREES OF AUXILIARY PLANE
REAL  DD2 (OUTPUT) DIP DIRECTIONS IN DEGREES OF AUXILIARY PLANE
REAL  SA2 (OUTPUT) SLIP ANGLE IN DEGREES OF AUXILIARY PLANE

DOUBLE PRECISION BOT
DOUBLE PRECISION DEL1
LOGICAL  FIRST
DOUBLE PRECISION PHI1
DOUBLE PRECISION PHI2
DOUBLE PRECISION RAD
DOUBLE PRECISION SGN
DOUBLE PRECISION TOP
DOUBLE PRECISION XLI
DOUBLE PRECISION XLMZ

DATA FIRST .TRUE./
SAVE FIRST, RAD

IF (FIRST) THEN
  FIRST = .FALSE.
  RAD = DATAN(1.0D0)/45.0D0
END IF

PHI1 = DD1 - 90.0D0
IF (PHI1 .LT. 6.0D0) PHI1 = PHI1 + 360.0D0
PHI1 = PHI1*RAD
DEL1 = DA1*RAD
SGN = SA1
XLI = SA1*RAD
C
TOP = DCOS(XLAM1)*DSIN(PHI1) - DCOS(DEL1)*DSIN(XLAM1)*DCOS(PHI1)
BOT = DCOS(XLAM1)*DCOS(PHI1) + DCOS(DEL1)*DSIN(XLAM1)*DSIN(PHI1)
DD2 = DATAN2(TOP, BOT)/RAD
PHI2 = (DD2 - 90.0D0)*RAD
IF (SA1 .LT. 0.00002)
DD2 = DD2 - 180.0D0
ENDIF
IF (DD2 .LT. 0.00002) DD2 = DD2 + 360.0D0
IF (DD2 .GT. 360.0D0) DD2 = DD2 - 360.0D0
DA2 = DACOS(DSIN(DABS(XLAM1))/DSIN(DEL1))/RAD
XLAM2 = -DCOS(PHI2)*DSIN(DEL1)*DSIN(PHI1) +
& DSIN(PHI2)*DSIN(DEL1)*DCOS(PHI1)
MACHINE ACCURACY PROBLEM
IF (ABS(XLAM2) .GT. 1.0D0) THEN
XLAM2 = DSIGN(1.0D0, XLAM2)
ENDIF
XLAM2 = DSIGN(DACOS(XLAM2), SGN)
SA2 = XLAM2/RAD
RETURN
END

C
C
C
C
C
SUBROUTINE HHOG (EUNIT, JSTRT, NSTRT, MSTRT, IGOOD, GFIT, NG,
& IDST, NDST, MXDIP, MXSLNS, MXSTRK, MXRAKE, NDELC, NPHIC, NLAMC)
C
C PERFORMS A "HEDGEHOG" SEARCH THROUGH COARSE SOLUTIONS WITH FITS LESS THAN FITLIM, IDENTIFIES SOLUTIONS BELONGING TO
C DISCRETE LOCALIZED MINIMA, AND RETURNS STRIKE, DIP, AND RAKE INDICES OF SOLUTION WITH BEST FIT WITHIN EACH MINIMA
C
INTEGER MxHOG 1 MAXIMUM NUMBER OF SOLUTIONS PER LOCALIZED MINIMA
PARAMETER (MxHOG = 200)

INTEGER EUNIT (INPUT) LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
REAL GFIT(MXDIP*MXSTRK*MXRAKE) (INPUT) CONTAINS FITS OF SOLUTIONS IN IGOOD
INTEGER IDST(MXSLNS, 3) (OUTPUT INDICES OF BEST FITTING SOLUTIONS IN EACH LOCALIZED MINIMA
INTEGER IGOOD(MXDIP*MXSTRK*MXRAKE, 4) (INPUT) INDICES OF SOLUTIONS WITH "GOOD" FITS DETERMINED BY COARSE SEARCH.
INTEGER JSTRT (INPUT) DIP INDEX OF BEST SOLUTION FROM COARSE SEARCH
INTEGER MSTRT (INPUT) RAKE INDEX OF BEST SOLUTION FROM COARSE SEARCH
INTEGER MXDIP (INPUT) MAXIMUM NUMBER OF DIP VALUES IN SEARCH
INTEGER MXRAKE (INPUT) MAXIMUM NUMBER OF RAKE VALUES IN SEARCH
INTEGER MXSLNS (INPUT) MAXIMUM NUMBER OR MULTIPLE SOLUTIONS PERMITTED
INTEGER MXSTRK (INPUT) MAXIMUM NUMBER OR STRIKE VALUES IN SEARCH
INTEGER NDST (OUTPUT NUMBER OF SOLUTIONS IN IDST
INTEGER NG (INPUT) NUMBER OF SOLUTIONS IN IGOOD
INTEGER NLAMC (INPUT) NUMBER OF INCREMENTS OF RAKE IN COARSE SEARCH
INTEGER NPHIC (INPUT) NUMBER OF INCREMENTS OF STRIKE IN COARSE SEARCH
INTEGER NSTRT (INPUT) STRIKE INDEX OF BEST SOLUTION FROM COARSE SEARCH
REAL FMIN (SMALLEST FIT VALUE WITHIN SET OF SOLUTIONS COMPRISING A LOCALIZED MINIMA
INTEGER IC (NUMBER OF SOLUTIONS USED AS CENTER POINT FOR EXPANSION
INTEGER ICACH(MxHOG) (POINTER ARRAY INDICES OF IGOOD
INTEGER ICT (TOTAL NUMBER OF SOLUTIONS IN A HEDGEHOG
INTEGER IG (INDEX OVER IGOOD
INTEGER IK (LOOP INDEX OVER ICACH
INTEGER J (DIP INDEX OF CENTER POINT FOR EXPANSION
INTEGER JJ (DIP INDEX OF NEARBY SOLUTION TO CENTERPOINT
INTEGER JJK (DIP INDEX OF NEARBY SOLUTION TO CENTERPOINT

C
C
C
C
C
INTEGER M0
INTEGER MM
INTEGER MMX
INTEGER N0
INTEGER NHH
INTEGER NN
INTEGER NNX

C
NHH - I
ICT = 0
J0 = JSTRT
M0 = MSTRT

C EXPAND ABOUT (J0, N0, M0) FOR NEAREST NEIGHBORS

20 DO 90 JJ = J0 - 1, J0 + 1
    IF (JJ .LE. #) THEN
        JJX = NDELC - JJ
    ELSE IF (JJ .GT. NDELC) THEN
        JJX = JJ - NDELC
    ELSE
        JJX = JJ
    END IF
    DO 80 NN = J0 - 1, J0 + 1
        IF (NN .LE. 0) THEN
            NNX = NPHIC - NN
        ELSE IF (NN .GT. NPHIC) THEN
            NNX = NN - NPHIC
        ELSE
            NNX = NN
        END IF
        DO 70 MM = M0 - 1, M0 + 1
            IF (MM .LE. #) THEN
                MMX = NLAMC - MM
            ELSE IF (MM .GT. NLAMC) THEN
                MMX = MM - NLAMC
            ELSE
                MMX = MM
            END IF
            C LOOK UP EACH SOLUTION IN IGOOD. IF FOUND, ANNOTATE IT WITH THE CURRENT VALUE OF NHH
            DO 60 IG = 1, NG
                IF (IGOOD(IG, 1) .EQ. JJX .AND. IGOOD(IG, 2) .EQ. NNX .AND. IGOOD(IG, 3) .EQ. MMX .AND. IGOOD(IG, 4) .EQ. 0) THEN
                    IGOOD(IG, 4) = NHH
                END IF
            C CHECK TO SEE IF THIS SOLUTION IS ALREADY IN A CACHE
            IF ICT .GT. M0 THEN
                DO 50 IK = 1, ICT
                    IF (ICACH(IK) .EQ. IG) GOTO 70
                50 CONTINUE
                END IF
            C STORE THIS SOLUTION IN CACHE
            ICT = ICT + 1
            IF (ICT .GT. M0+1) THEN
                WRITE (UNIT, *) 'ERROR: NUMBER OF SOLUTIONS WITHIN HEDGEHOG EXCEEDS ', M0+1
                STOP
            END IF
90 CONTINUE
C
C
C
END IF
ICACH(ICT) = IG
END IF
60 CONTINUE
70 CONTINUE
80 CONTINUE
90 CONTINUE
C
C SELECT NEXT SOLUTION WITHIN CURRENT HEDGEHOG AS STARTING POINT FOR EXPANSION
C
IC = IC + 1
IF (IC .LE. ICT) THEN
J# = IGOOD(ICACH(IC), 1)
N# = IGOOD(ICACH(IC), 2)
M# = IGOOD(ICACH(IC), 3)
GOTO 2#
ELSE
C
C FINISHED PROCESSING CACHE FOR CURRENT HEDGEHOG
C
ICT = #
IC = #
NHH = NHH + 1
IF (NHH .GT. MXSLNS) THEN
PRINT *, '***** HHOG ERROR: NUMBER OF MULTIPLE SOLUTIONS EXCEEDS ', MXSLNS, ' *****'
STOP
END IF
C
C GET NEXT SOLUTION FROM IGOOD THAT DOES NOT ALREADY BELONG TO A MINIMA
C
DO 100 IG = 1, NG
IF (IGOOD(IG, 4) .EQ. 0) THEN
IG = IGOOD(IG, 1)
N0 = IGOOD(IG, 2)
M0 = IGOOD(IG, 3)
GOTO 2#
END IF
100 CONTINUE
END IF
C
C IDENTIFY SOLUTION CORRESPONDING TO FIT MINIMUM WITHIN EACH HEDGEHOG
C
DO 130 NDST = 1, NHH - 1
FMIN = 1.
DO 120 IG = 1, NG
IF (IGOOD(IG, 4) .EQ. NDST .AND. GFIT(IG) .LE. FMIN) THEN
IDST(NDST, 1) = IGOOD(IG, 1)
IDST(NDST, 2) = IGOOD(IG, 2)
IDST(NDST, 3) = IGOOD(IG, 3)
FMIN = GFIT(IG)
END IF
120 CONTINUE
130 CONTINUE
NDST = NHH - 1
C
RETURN
END
C
C
FUNCTION RDIFF (RAKE1, RAKE2)
C RETURNS WITH THE SMALLEST ABSOLUTE DIFFERENCE IN SLIP ANGLE BETWEEN RAKE1 AND RAKE2.
C RAKE CONVENTION FOLLOWS AKI & RICHARDS, 1980, QUANTITATIVE SEISMOLOGY, P. 114
C
REAL RAKE1 ! (INPUT) FIRST RAKE
REAL RAKE2 ! (INPUT) SECOND RAKE
REAL A ! STORES FIRST RAKE
REAL B ! STORES SECOND RAKE
REAL C ! STORES RAKE DIFFERENCE

C
RDIF - 999.
A = RAKE1
IF (RAKE1 .LT. 0.) A = 360. + RAKE1
B = RAKE2
IF (RAKE2 .LT. 0.) B = 360. + RAKE2
C = ABS(A - B)
IF (C .GT. 180.) C = 360. - C
RDIF = C
RETURN
END
PROGRAM FPPLOT

VERSION 1.0 - OCTOBER 31, 1985

PURPOSE: PLOT EARTHQUAKE RAY POLARITIES AND FAULT PLANES ON A LOWER HEMISPHERE EQUAL AREA PROJECTION.


REQUIRED ROUTINES: CALCOMP STYLE PLOT ROUTINES PLOTS, PLOT, NEWPEN, SYMBOL.

DEPARTURES FROM FORTRAN-77 STANDARD:
1. KEYWORD "READONLY" OPEN STATEMENT
2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS
3. ARGUMENT LIST BUILT-IN FUNCTION "XREF"
4. CALLS TO VAX SYSTEM ROUTINES "SYSSINIT", "SYSSCHDWK", "SYSSHIBER" (SUBROUTINE DELAY)

OUTPUT: GRAPHIC OUTPUT ONLY

AUTHORS: PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK. SOME OF THE ROUTINES WERE ADOPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.

REAL AIN ! RAY ANGLE OF INCIDENCE
REAL ANS ! FLAG: Y-N=DO (NOT) PLOT STATION NAMES ADJACENT TO FIRST MOTIONS
REAL AZM ! X POSITION OF LARGE CIRCLE CENTER
REAL CXI ! X POSITION OF SMALL CIRCLE CENTER
REAL CX2 ! Y POSITION OF LARGE CIRCLE CENTER
REAL CV1 ! Y POSITION OF SMALL CIRCLE CENTER
REAL CV2 ! DIP ANGLE OF NEARBY SOLUTIONS
REAL DA ! DIP DIRECTION OF NEARBY SOLUTIONS
REAL DNI(11) ! DIP DIRECTION OF NEARBY SOLUTIONS
REAL DD ! DIP DIRECTION OF NEARBY SOLUTIONS
REAL DNI(11) ! DIP DIRECTION OF NEARBY SOLUTIONS
REAL DISC ! DIP ANGLE OF NEARBY SOLUTIONS
REAL DIST ! FIRST MOTION DISCREPANT WITH SOLUTION
REAL FMT71 SUMMARY CARD
REAL FILNAM ! FILE NAME OF DATA
REAL EVENT ! FIRST MOTION DESCRIPTION (EG. IPUI8)
REAL HITE1 ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
REAL HITE2 ! LEFTMOST X POSITION OF TITLE, SYMBOL LEGEND
REAL HITE3 ! LEFTMOST X POSITION OF FIRST MOTION LEGEND
REAL I ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER INDX ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER I/O STATUS DESCRIPTOR
INTEGER LINE ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NAME ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NEV ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NLINE ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NLIN ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NREAD ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
INTEGER NSKIP ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
REAL P ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
CHARACTER*1 PRMK ! EVENT NUMBER
CHARACTER*4 NAME ! EVENT NUMBER
CHARACTER*8 LINE ! EVENT NUMBER
CHARACTER*4 NAME ! EVENT NUMBER
CHARACTER*4 NAME ! EVENT NUMBER
CHARACTER*1 SYN ! EVENT NUMBER
CHARACTER*8 TITLE ! EVENT NUMBER
REAL PT ! EVENT NUMBER
REAL RT ! EVENT NUMBER
REAL XPOS1 ! EVENT NUMBER
REAL XPOS2 ! EVENT NUMBER
REAL XPOS3 ! EVENT NUMBER
REAL XPOS1 ! EVENT NUMBER
REAL XPOS2 ! EVENT NUMBER
REAL XPOS3 ! EVENT NUMBER
REAL YPOS1 ! EVENT NUMBER
REAL YPOS2 ! EVENT NUMBER
REAL YPOS3 ! EVENT NUMBER
PARAMETER (CX1 = 3.8, CX2 = 7.25, CY1 = 3.8, CY2 = 1.25)
PARAMETER (HITE1 = 0.2, HITE2 = 0.07, HITE3 = 0.1, RMAX1 = 2.95)
PARAMETER (RMAX2 = 1.25, XPOS1 = 0.1, XPOS2 = 6.5)
PARAMETER (YPOS1 = 7.3, YPOS2 = 0.1, YPOS3 = 6.0)
PI = ATAN(1.0)*4.0
RAD = PI/180.0

C 10 WRITE (6, 15) 'ENTER NAME OF DATA FILE: '
15 FORMAT ('"'. A)
READ (5, 16, ERR = 10) FILNAM
16 FORMAT (A)
OPEN (UNIT = 2, FILE = FILNAM, ERR = 10, STATUS = 'OLD', BLANK = 'ZERO', READONLY)
WRITE (6, 15) 'PLOT STATION NAMES (Y OR N)? '
READ (5, 16, IOSTAT = IDS) ANS
IF ((ANS .NE. 'Y' .AND. ANS .NE. 'N') .OR. IDS .NE. 0) THEN
   PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN *****'
   GOTO 20
END IF
WRITE (6, 15) 'ENTER NUMBER OF MECHANISMS TO SKIP (INCLUDING MULTIPLLE SOLUTIONS): '
25 READ (5, *, IOSTAT = IOS) NSKIP
IF (NSKIP .LT. 0 .OR. IOS .NE. 0) THEN
   PRINT *, '***** INVALID NUMBER; TRY AGAIN *****'
   GOTO 25
END IF
C
C READ HYP071 HEADER CARD (FIRST LINE IN MODEL FILE)
C
NEV = 0
ILINE = 1
READ (2, 30, ERR = 2000) TITLE
30 FORMAT (A)
INDX = 1
DO 40 I = 1, 80
   IF (TITLE(I:1) .NE. ' ') THEN
      INDX = I
      GOTO 50
   END IF
40 CONTINUE
50 TITLE = TITLE(INDX:LEN(TITLE))
C
C INITIALIZE PLOT PROGRAM
C
CALL PLOTS (0., 0., 0)
C
C READ EVENT
C
60 ILINE = ILINE + 1
READ (2, 70, END = 1000, ERR = 2000) EVENT
70 FORMAT (IX, A132)
NEV = NEV + 1
IF (NEV .GT. NSKIP) THEN
   C
   C START NEW FRAME
   C
   CALL ERASE
   CALL DELAY ('0000 00i00i01.00')
   CALL PLOT (1., 1., -3 >
   FIRST = .TRUE.
   CALL NEWPEN (1)
   READ (EVENT, 75) DD1, DA I, SA1
75 FORMAT (T81, F4.0, F3.0, F4.0)
   C
   C PLOT SUMMARY CARD & EXPLANATION OF SYMBOLS
   C
   YPOS = YPOS1
   CALL SYMBOL (XPOS1, YPOS, HITE3, XREF(TITLE), 0., LEN(TITLE))
   YPOS = YPOS - 3
   CALL SYMBOL (XPOS1, YPOS, HITE3, XREF(EVENT(I:80>>, 0., 80)
   YPOS = YPOS - 3
   C
   C END OF EVENT
   C
   GOTO 60
END IF
CALL SYMBOL(XPOS1, YPOS, HITE3, XREF(EVENT(82,132), 0, 51))
CALL NEWPEN(1)
YPOS = YPOS2
CALL CIRCLE(HITE2, 2.0*PI, XPOS1, YPOS + HITE2/2.)
LINE(1:1:1) = 'COMPRESSION'
CALL SYMBOL(XPOS1 + .2, YPOS, HITE2, XREF(LINE), 0, 11)
YPOS = YPOS + 2.0*HITE2
CALL TRINGL(HITE2, XPOS1, YPOS + HITE2/2.)
LINE(1:1:1) = 'DILATATION'
CALL SYMBOL(XPOS1 + .2, YPOS, HITE2, XREF(LINE), 0, 11)
C
C PLOT BIG & LITTLE STEREO NET PERIMETERS
C
CALL STRNET(CXI, CY1, RAD, RMAX1)
CALL STRNET(CX2, CY2, RAD, RMAX2)
C
C PLOT NODAL PLANES
C
CALL PLOTPL(CXI, CY1, DA1, PI, RAD, RMAX1, DD2 - 90.)
CALL AUXPLN(001, DA1, SA1, DD2, OA2, SA2)
CALL PLOTPL(CXI, CY1, DA2, PI, RAD, RMAX1, DD2 - 90.)
C
C PLOT 'P' AND 'T' AXES IN BIG NET
C
NAME = ' '  
WT = 1.0
CALL NEWPEN(3)
CALL TTPLOT(CXI, CY1, DA1, DD2, HITE1, PI, RAD, RMAX1, SA1, WT)
C
C PLOT THE P AND T AXES IN SMALL NET CORRESPONDING TO THE SET OF 'NEIGHBORING SOLUTIONS'
C
CALL TTPLOT(CX2, CY2, DA1, DD2, HITE2, PI, RAD, RMAX2, SA1,WT)
CALL NEWPEN(1)
END IF
LINE = LINE + 1
READ (2, *, ERR = 2000) NSTAR
NL1NE = (NSTAR-1)/11 * 1  
DO 90 I = 1, NL1NE
IF (I .LT. NL1NE) THEN
NREAD = II
ELSE
NREAD = MOD(INSTAR, 11)
END IF
IF (NAME .NE. '"') THEN
C
C REPORT DISCREPANT OBSERVATIONS
C
IF (NEV .GT. NSK1P) THEN
IF (DISC .EQ. '"') THEN
IF (FIRST) THEN
LINE(1:2:3) = 'DISCREPANT OBSERVATIONS'
YPOS = YPOS3
CALL SYMBOL(XPOS2, YPOS, HITE2, XREF(LINE), 0, 23)
YPOS = YPOS - HITE2*.5
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
C
C CALL ABD
C
C CALL BDS
C
C CALL PLOT
LINE(1:25) = 'STAT DIST AZM AIN PRMK'
CALL SYMBOL (XPOS2, YPOS, HITE2, XREF(LINE), 0., 25)
YPOS = YPOS - HITE2*1.5
CALL SYMBOL (XPOS2, YPOS, HITE2, XREF(LINE), 0., 25)
YPOS = YPOS - HITE2*1.5
FIRST = .FALSE.
END IF
WRITE (LINE, 120) NAME, DIST, NINT(AZM), NINT(AIN), PRMK
FORMAT (A4, F5.1, 215, 2X, A4, 15X)
CALL SYMBOL (XPOS2, YPOS, HITE2, XREF(LINE), 0., 31)
YPOS = YPOS - HITE2*1.5
C PLOT FIRST MOTION
C END IF
IF (ANS .EQ. 'N') NAME = ' '
IF (PRMK(3:3) .EQ. 'U' .OR. PRMK(3:3) .EQ. '+') THEN
SYM = 'C'
ELSE
SYM = 'D'
END IF
CALL PLTSYM (AIN, AZM, CX1, CY1, HITE1, NAME, PI, RAD, & RMAX1, SYM, WT)
END IF
GOTO 100
END IF
GOTO 60
C END OF FILE
C 1000 CALL PLOT (0., 0., 999)
CLOSE (2)
STOP
C READ ERROR
C 2000 PRINT *, 'READ ERROR ON LINE ', ILINE
STOP
END
C SUBROUTINE ERASE
C SENDS A SCREEN ERASE CODE TO A TEKTRONIX TERMINAL
C CHARACTER*1 A
CHARACTER*1 B
A = CHAR(27)
B = CHAR(12)
WRITE (6, 10) A, B
10 FORMAT (1X, 2A1)
RETURN
END
SUBROUTINE DELAY (TIME)

C CREATES A DELAY FOR SCREEN RECOVERY

CHARACTER(*) TIME

DOUBLE PRECISION B

C CONVERT ASCII TIME TO BINARY TIME

CALL SYSSBINTIM (TIME,B)

C SCHEDULE A WAKEUP FOR A ELAPSED TIME (NEGATIVE B)

CALL SYSSSCHDWK (,,-B,)

C HIBERNATE AND REAWAKE

CALL SYS$HIBER

RETURN

END

SUBROUTINE CIRCLE (SIZE, TWOPI, X0, Y0)

C PLOT A CIRCLE

REAL SIZE

REAL TWOPI

REAL X0

REAL Y0

C

REAL ANGLE

INTEGER J

INTEGER N

REAL SIZE2

REAL X

REAL Y

C

SIZE2 = SIZE*0.5

C COMPUTE OPTIMUM # OF POINTS TO DRAW

N = 2*SQRT(SIZE2*20.)

IF (N .LT. 10) N = 10

C DRAW CIRCLE

X = X0 + SIZE2

CALL PLOT (X, Y0, 3)

DO 10 J = 1, N

ANGLE = TWOPI*FLOAT(J)/FLOAT(N)

X = X0 + SIZE2*COS(ANGLE)

Y = Y0 + SIZE2*SIN(ANGLE)

CALL PLOT (X, Y, 2)

10 CONTINUE

RETURN

END
SUBROUTINE TRINGL (SIZE, X#, Y#)

C PLOT A TRIANGLE

REAL SIZE  ! SIZE OF TRIANGLE
REAL X#  ! X POSITION OF CENTER
REAL Y#  ! Y POSITION OF CENTER

REAL SIZEX  ! SCRATCH VARIABLE
REAL SIZEX  ! SCRATCH VARIABLE
REAL X  ! X PLOT POSITION
REAL Y  ! Y PLOT POSITION

PARAMETER (COS30 = .8660254, SIN30 = .5)

SIZEY = SIZE*SIN30
SIZEX = SIZE*COS30

MOVE TO TOP

Y = Y# + SIZEY
CALL PLOT (X#, Y, 3)

DRAW TO LOWER LEFT

X = X# - SIZEX
Y = Y# - SIZEY/2
CALL PLOT (X, Y, 2)

DRAW TO LOWER RIGHT

X = X# + SIZEX
CALL PLOT (X, Y, 2)

DRAW TO TOP

Y = Y# + SIZEY
CALL PLOT (X#, Y, 2)

RETURN

END

SUBROUTINE STRNET (CX, CY, RAD, RMAX)

C PLOT PERIMETER OF A STEREO NET

REAL CX  ! X POSITION OF CIRCLE CENTER
REAL CY  ! Y POSITION OF CIRCLE CENTER
REAL RAD  ! PI/180
REAL RMAX  ! RADIUS OF CIRCLE

REAL CSIZ  ! SCRATCH VARIABLE (RMAX/180)
INTEGER I  ! LOOP INDEX OVER DEGREES
INTEGER N  ! TESTS 18 DEGREE TICK POSITION
INTEGER NN  ! TESTS 90 DEGREE TICK POSITION
REAL P  ! TICK LENGTH
REAL PHI 1 AZIMUTH IN RADIANS
REALX 1 X POSITION OF CIRCLE
REAL XP 1 X POSITION OF END OF TICK
REALY 1 Y POSITION OF CIRCLE
REAL YP 1 Y POSITION OF END OF TICK

C CALL NEWPEN (2)
C DRAW CIRCLE @ 1 DEGREE INCREMENTS
C
DO 10I = 1.361
PHI = (I - 1)*RAD
X = RMAX*COS(PHI) + CX
Y = RMAX*SIN(PHI) + CY
N = (I - 1) - ((I - 1)/10)*10
NN = (I - 1) - ((I - 1)/90)*90
P = .01*RMAX
IF (N .EQ. 0) .AND. (I .GT. 10) P = .02*RMAX
IF (NN .EQ. 0) .AND. (I .GT. 90) P = .04*RMAX
XP = (RMAX + P*COS(PHI) + CX
YP = (RMAX + P*SIN(PHI) + CY
IF (I .GT. I) THEN
CALL PLOT (X, Y, 2)
CALL PLOT (XP, YP, 2)
CALL PLOT (X, Y, 2)
ELSE
CALL PLOT (X, Y, 3)
END IF
10 CONTINUE
C PLOT AT CENTER
CSIZ = .01*RMAX
CALL PLOT (CX - CSIZ, CY, 3)
CALL PLOT (CX + CSIZ, CY, 2)
CALL PLOT (CX, CY - CSIZ, 3)
CALL PLOT (CX, CY + CSIZ, 2)
CALL NEWPEN (1)
C RETURN
END
C
C SUBROUTINE PLOTPL (CX, CY, D PIDG, PI, RAD, RMAX, STRKD G)
C PLOTS FAULT PLANE ON LOWER HEMISPHERE STEREO NET
C
REAL CX 1 X POSITION OF CIRCLE CENTER
REAL CY 1 Y POSITION OF CIRCLE CENTER
REAL D PIDG 1 DIP ANGLE IN DEGREES
REAL PI 1 PI
REAL RAD 1 PI/180
REAL RMAX 1 RADIUS OF CIRCLE
REAL STRKD G 1 STRIKE ANGLE IN DEGREES
REAL ANG 1 ANGLE IN RADIANS
REAL A INP(91) 1 ANGLE OF INCIDENCE IN RADIANS
REAL ARG 1 DUMMY ARGUMENT
REAL AZ 1 AZIMUTH
REAL CON 1 RADIUS COEFFICIENT
REAL DIPRD 1 DIP ANGLE IN RADIANS
INTEGER I
INTEGER MI
REAL RADIUS
REAL SAZ(91)
REAL STRKRD
REAL TAZ
REAL TPD
REAL X
REAL Y
C STRKRD = STRKRD*RAD
DIPRD = DIPIDG*RAD
TPD = TAN(PI*.5 - DIPRD)**2
C CASE OF VERTICAL PLANE
C IF (DIPIDG .EQ. 90.0) THEN
X = RMAX*SIN(STRKRD) * CX
Y = RMAX*COS(STRKRD) * CY
CALL PLOT (X, Y, 3)
X = RMAX*S1N(STRKRD * PI) + CX
Y = RMAX*COS(STRKRD * PI) + CY
CALL PLOT (X, Y, 2)
RETURN
END IF
C COMPUTE ANGLE OF INCIDENCE, AZIMUTH
C DO 10 I = 1, 90
ANG = FLOAT(I - 1)*RAD
ARG = SQRT((COS(DIPRD)**2)*COS(ANG)**2))/COS(ANG)
SAZ(I) = ATAN(ARG)
TAZ = TAN(SAZ(I))**2
ARG = SQRT(TPD + TPD*TAZ * TAZ)
AINP(I) = ACOS(TAN(SAZ(I))/ARG)
10 CONTINUE
SAZ(91) = 90.*RAD
AINP(91) = PI*.5 - DIPRD
C PLOT PLANE
C CDN = RMAX*SORT(2.)
DO 20 I = 1, 180
IF (I .LE. 90) THEN
MI = I
MI = 181 - I
END IF
AZ = PI - SAZ(MI) + STRKRD
RADIUS = CDN*SIN(AINV(MI)*.5)
X = RADIUS*SIN(AZ) + CX
Y = RADIUS*COS(AZ) + CY
IF (I .EQ. 1) THEN
CALL PLOT (X, Y, 3)
ELSE
CALL PLOT (X, Y, 2)
ENDIF
20 CONTINUE
RETURN
END
SUBROUTINE AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
CALCULATE THE AUXILLIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.
WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)
ANGLE VARIABLES PHI, DEL AND LAM ARE AS DEFINED IN AKI AND RICHARDS, (1900), P.114.

DIP ANGLE IN DEGREES
DIP DIRECTIONS IN DEGREES
SLIP ANGLE IN DEGREES
DIP ANGLE OF AUXILLIARY PLANE
DIP DIRECTION OF AUXILLIARY PLANE
SLIP ANGLE OF AUXILLIARY PLANE
SCRATCH VARIABLE
DIP ANGLE OF PRINCIPAL PLANE IN RADIANS
TEST: TRUE IF FIRST TIME INTO ROUTINE
FAULT PLANE STRIKE OF PRINCIPAL PLANE
CONVERSION FACTOR FROM DEGREES TO RADIANS
Saves principal plane slip angle for assigning proper sign to auxiliary
SCRATCH VARIABLE
SLIP ANGLE OF PRINCIPAL PLANE IN RADIANS
SLIP ANGLE OF AUXILLIARY PLANE

DATA FIRST /.TRUE./
SAVE FIRST, RAD
IF (FIRST) THEN
FIRST = .FALSE.
RAD = DATAN1(.00/)/45.00
ENDIF

PHI1 = DD1 - 90.00
IF (PHI1 .LT. 0.00) PHI1 = PHI1 + 360.00
PHI1 = PHI1*RAD
DELI = DA1*RAD
SGN = SA1
XLAM1 = SA1*RAD

TOP = DCON(XLAM1)*DSIN(PHI1) - DCON(DELI)*DSIN(XLAM1)*DCOS(PHI1)
BOT = DCON(XLAM1)*DCOS(PHI1) + DCON(DELI)*DSIN(XLAM1)*DSIN(PHI1)
DD2 = DATAN2(TOP, BOT)/RAD
PHI2 = (DD2 - 90.00)*RAD
IF (SA1 .LT. 0.00) DD2 = DD2 + 360.00
IF (DD2 .GT. 90.00) DD2 = DD2 - 360.00

DA2 = DCON(SIN(DABS(XLAM1)))*DSIN(DELI)/RAD
XLAM2 = -DCON(PHI2)*DSIN(DELI)*DSIN(PHI1) + & DSIN(PHI2)*DSIN(DELI)*DCCOS(PHI1)

DOUBLE PRECISION BOT
DOUBLE PRECISION DELI
LOGICAL FIRST
DOUBLE PRECISION PHI1
DOUBLE PRECISION PHII
DOUBLE PRECISION RAD
DOUBLE PRECISION SGN
DOUBLE PRECISION TOP
DOUBLE PRECISION XLAM1
DOUBLE PRECISION XLAM2

C MACHINE ACCURACY PROBLEM
C
IF (XLAM2 .GT. 1.00) THEN
XLAM2 = 1.00
ENDIF
XLAM2 = DSIGN(DCONS(XLAM2), SGN)
SA2 = XLAM2/RAD

C
SUBROUTINE TPPLOT (CX, CY, DA1, DD1, HITE, PI, RAD, RMAX, SA1, WT)
C
C PLOT P AND T AXES
C
REAL CX  ! X POSITION OF CIRCLE CENTER
REAL CY  ! Y POSITION OF CIRCLE CENTER
REAL DA1 ! DIP DIRECTION OF PRINCIPLE PLANE
REAL DD1 ! DIP ANGLE OF PRINCIPLE PLANE
REAL HITE ! HEIGHT OF P,T SYMBOL
REAL PI  ! PI
REAL RAD ! PI/180
REAL RMAX ! RADIUS OF CIRCLE
REAL SA1 ! RAKE OF PRINCIPLE PLANE
REAL WT  ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
C
REAL AIN1 ! ANGLE OF INCIDENCE OF P/T AXIS
REAL AIN2 ! ANGLE OF INCIDENCE OF T/P AXIS
REAL ANG ! ANGLE OF PLOT SYMBOL
REAL AZ1 ! AZIMUTH OF P/T AXIS
REAL AZ2 ! AZIMUTH OF T/P AXIS
CHARACTER*4 BLANK ! BLANK
CHARACTER*8 DA2 ! DIP ANGLE OF AUXILIARY PLANE
CHARACTER*8 DD2 ! DIP DIRECTION OF AUXILIARY PLANE
CHARACTER*1 SYM1 ! P/T PLOT SYMBOL
CHARACTER*1 SYM2 ! T/P PLOT SYMBOL
C
PARAMETER (ANG = 8.8, BLANK = ' ')
C
C FIND AUXILIARY PLANE
C
CALL AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C FIND P AND T AXES
C
CALL TANDP (AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI, RAD)
IF (SA1 .LT. 0.) THEN
   SYM1 = 'P'
   SYM2 = 'T'
ELSE
   SYM1 = 'T'
   SYM2 = 'P'
END IF
C
C PLOT SYMBOLS
C
CALL PLTSYM (AIN1, AZ1, CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM1, & WT)
CALL PLTSYM (AIN2, AZ2, CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM2, & WT)
C
RETURN
END
SUBROUTINE TANDP(AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI, RAD)

GIVEN TWO PLANES COMPUTE AZ AND ANGLE OF INCIDENCE OF P & T AXES

REAL AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI, RAD

REAL ALAT1, ALON1, ALAT2, ALON2, AZIMTH, AZ0, PLUNGE, DELTA, SHIF1, XPOS, YPOS

PARAMETER (SHIFT = 0.7853981)

ALAT1 = (90. - DA1)*RAD
ALON1 = DD1*RAD
ALAT2 = (90. - DA2)*RAD
ALON2 = DD2*RAD
CALL REFPT (ALAT2, ALON2)
CALL DELAZ (ALAT1, ALON1, DELTA, AZ0, BAZM, XPOS, YPOS)
CALL BACK (SHIFT, AZ0, PLUNGE, AZIMTH)

IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.*PI, AZIMTH)
AZ1 = AZIMTH/RAD
AZ0 = AZ0 + PI
AZIMTH = AZ0 + PI
CALL BACK (SHIFT, AZ0, PLUNGE, AZIMTH)

RETURN

END

SUBROUTINE GEOCEN

GEOCEN - CALCULATE GEOCENTRIC POSTITIONS, DISTANCES, AND AZIMUTHS (BRUCE JULIAN, USGS MENLO PARK, CA)

THE GEOCENTRIC DISTANCE DELTA AND AZIMUTH AZ0 FROM POINT (LAT0, LON0) TO POINT (LAT1, LON1) ARE CALCULATED FROM

\[
\cos(\Delta) = \cos(LAT0') \cdot \cos(LAT1') \cdot \cos(LON1 - LON0) + \sin(LAT0') \cdot \sin(LAT1')
\]

\[
\sin(\Delta) = \sin(LAT0') \cdot \sin(LAT1') \cdot \cos(LON1 - LON0) + \sin(LAT0') \cdot \sin(LAT1') \cdot \cos(LON1 - LON0)
\]

\[
\tan(\Delta) = \sin(LAT0') \cdot \sin(LAT1') \cdot \sin(LON1 - LON0) / \sin(LAT0') \cdot \sin(LAT1') \cdot \cos(LON1 - LON0)
\]

WHERE

\[
LAT0' = 90. - LAT0
\]

\[
LON0' = 90. - LON0
\]

\[
LAT1' = 90. - LAT1
\]

\[
LON1' = 90. - LON1
\]

\[
\Delta = \Delta - 90.
\]

\[
\Delta = \Delta + 90.
\]
A = COs(LAT1')*SiN(LON1-LON0)
B = COs(LAT0')*SiN(LAT1') - SiN(LAT0')*COs(LAT1')*COs(LON1-LON0)
LAT0', LAT1' = GEOCENTRIC LATITUDES OF POINTS
LON0, LON1 = LONGITUDES OF POINTS

THE GEOCENTRIC LATITUDE LAT' IS GOTTEN FROM THE GEOGRAPHIC LATITUDE LAT BY TAN(LAT') = (1 - ALPHA)*(1 - ALPHA)*TAN(LAT), WHERE ALPHA IS THE FLATTENING OF THE ELLIPSOID. SEE FUNCTION GGTOGC FOR CONVERSION.

THE BACK AZIMUTH IS CALCULATED BY THE SAME FORMULAS WITH (LAT0', LON0) AND (LAT1', LON1) INTERCHANGED.

AZIMUTH IS MEASURED CLOCKWISE FROM NORTH THRU EAST.

REAL R, THETA 1 RADIUS, AZIMUTH IN POLAR COORDINATES
REAL AZ0 1 AZIMUTH FROM SECONDARY POINT TO REFERENCE POINT IN RADIANS
REAL AZ1 1 AZIMUTH FROM REFERENCE POINT TO SECONDARY POINT IN RADIANS
REAL CT0 1 SINE OF REFERENCE POINT LATITUDE
REAL CT1 1 SINE OF SECONDARY POINT LATITUDE
REAL ST0 1 COSINE OF AZIMUTH TO SECONDARY POINT
REAL ST1 1 COSINE OF AZIMUTH TO SECONDARY POINT
REAL DELTA 1 GEOCENTRIC DISTANCE IN DEGREES
REAL DOLON 1 AZIMUTH IN POLAR COORDINATES TO SECONDARY POINT
REAL FLAT 1 EQUATORIAL RADIUS (CHOVITZ, 1981, EOS, 62, 65-67)
REAL FLAT 1 EARTH FLATTENING CONSTANT (CHOVITZ, 1981, EOS, 62, 65-67)
REAL LAMBDA 1 DUMMY VARIABLE
REAL LAT 1 LATITUDE IN RADIANS
REAL LON 1 LONGITUDE IN RADIANS
REAL OLAT 1 ORIGIN LATITUDE IN RADIANS
REAL OLON 1 ORIGIN LONGITUDE IN RADIANS
REAL PI 1 3.14159...
REAL RADIUS 1 EARTH RADIUS AT COLAT
REAL SDELT 1 COSINE OF DELTA TO SECONDARY POINT
REAL SDLON 1 SINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
REAL ST 1 COSINE OF REFERENCE POINT LATITUDE
REAL ST 1 COSINE OF SECONDARY POINT LATITUDE
REAL TWOPI 1 2*PI

SAVE ST#, CT#, PHI#, OLAT
PARAMETER (PI = 3.1415926535897, TWOPI = 2.*PI)
PARAMETER (FLAT = 1./298.257, ERAD = 6378.137)
PARAMETER (LAMBDA = FLAT**2 - FLAT)/(1. - FLAT)**2

REFPT - STORE THE GEOCENTRIC COORDINATES OF THE REFERENCE POINT
ENTRY REFPT(OLAT, OLON)

ST# = COS(OLAT)
CT# = SIN(OLAT)
PHI# = OLON
RETURN

DELAZ - CALCULATE THE GEOCENTRIC DISTANCE, AZIMUTHS
ENTRY DELAZ(LAT, LON, DELTA, AZ0, AZ1, X, Y)

CT1 = SIN(LAT)
ST1 = COS(LAT)
IF ((CT1 - CT0) .EQ. 0. .AND. (LON - PHI0) .EQ. 0.) THEN
DELTA = 0.
AZ0 = 0.
AZ1 = 0.
ELSE
SDLON = SIN(LON - PHI0)
COLON = COS(LON - PHI0)

...
CALL CVRTOP (ST0*CT1 - ST1*CT0 + CDLON + CT0*CT1, ST1*SDLON + SDLON, SDLON, AZ0)
DELTA = ATAN2(SDELT, CDELT)
CALL CVRTOP (ST1*CT0 - ST0*CT1 - CDLON, CT0*CT1 + SDLON, SDLON, AZ1)
IF (AZ0 .LT. 0.0) AZ0 = AZ0 * TWOPI
IF (AZ1 .LT. 0.0) AZ1 = AZ1 * TWOPI
END IF

COLAT = PI/2. - (LAT * OLAT)/2.
RADIUS = ERAD/SQRT(1.0 + LAMBDA*COS(COLAT)**2)
V = RADIUS*DELTA*COS(AZ0)
X = RADIUS*DELTA*SIN(AZ0)
RETURN

C BACK - CALCULATE GEOCENTRIC COORDINATES OF SECONDARY POINT FROM DELTA, AZ
ENTRY BACK (DELTA, AZ0, LAT, LON)
SDELT = SIN(DELTA)
CDELT = COS(DELTA)
CT0 = COS(AZ0)
CT1 = ST0*SDELT*CT0 + CT0*CDELT
CALL CVRTOP (ST0*CDELT - CT0*SDELT*CT0, SDELT*SIN(AZ0), ST1, DLON)
LAT = ATAN2(CT1, ST1)
LON = PHI0 * DLON
IF (ABS(LON) .GT. PI) LON = LON - SIGN(TWOPI, LON)
RETURN
END

C SUBROUTINE CVRTOP(X, Y, R, THETA)
C CVRTOP - CONVERT FROM RECTANGULAR TO POLAR COORDINATES (BRUCE JULIAN, USGS MENLO PARK, CA)
REAL X, Y
REAL R, THETA
R = SQRT(X*X + Y*Y)
THETA = ATAN2(Y, X)
RETURN
END

C SUBROUTINE PLTSYM (AIN, AZ, CX, CY, HITE, NAME, PI, RAD, RMAX, & SYM, WT)
C PLOT EITHER FIRST MOTION SYMBOL (C,D,*,-> WITH STATION NAME NEXT TO SYMBOL, OR STRESS AXES SYMBOL <P & T)
REAL AIN ! ANGLE OF INCIDENCE OF SYMBOL
REAL AZ ! AZIMUTH OF SYMBOL
REAL CX ! X POSITION OF CIRCLE CENTER
REAL CY ! Y POSITION OF CIRCLE CENTER
REAL HITE
CHARACTER*4 NAME ! STRING TO BE PLOTTED TO RIGHT OF SYMBOL
REAL PI ! PI
REAL RAD ! RADIUS OF CIRCLE
REAL RMAX ! RADIUS OF CIRCLE
CHARACTER*1 SYM ! PLOT SYMBOL
REAL WT

REAL AINR
REAL ANG
REAL AZR
REAL CON
LOGICAL FIRST
REAL R
REAL SIZE
REAL SYMSIZ
REAL X
REAL Y

C PARAMETER (ANG = 0.0, SYMSIZ = 0.2)
C
AZR = AINR
AZR = AINR
C
C UPGOING RAYS
C
IF (AIN .GT. 90.) THEN
  AINR = PI - AINR
  AZR = PI - AZR
END IF
C
CON = RMAX*SORT(2.0)
R = CON*SIN(AINR*0.5)
X = R*SIN(AZR) + CX
Y = R*COS(AZR) + CY
C
C STRESS AXIS SYMBOL
C
IF ((SYM .EQ. 'P') .OR. (SYM .EQ. 'T')) THEN
  X = X - .286*HITE
  Y = Y - .5*HITE
  CALL SYMBOL (X, Y, HITE, XREF(SYM), ANG, 1)
ELSE
C
C FIRST MOTION SYMBOL
C
  SIZE = SYMSIZ*WT
  IF (AIN .GT. 90.) THEN
    CALL NEWPEN (4)
  ELSE
    CALL NEWPEN (1)
  END IF
  IF (SYM .EQ. 'C') THEN
    CALL CIRCLE (SIZE, 2.*PI, X, Y)
  ELSE
    CALL TRINGL (SIZE, X, Y)
  END IF
C
C PLOT STATION NAME
C
  IF (NAME .NE. ' ') CALL SYMBOL (X + SIZE/2., Y, SIZE/2.,
  & XREF(NAME), 0., 4)
C
RETURN
END
PROGRAM FPPAGE

PURPOSE: PLOT EARTHQUAKE RAY POLARITIES AND FAULT PLANES ON A LOWER HEMISPHERE EQUAL AREA PROJECTION.

Makes multiple plots per page.

INPUT FILE: A file of the type "*.pol", which is generated by the program "FPFIT" (see "FPFIT, FPPLT, and FPPAGE"
FORTAN COMPUTER PROGRAMS FOR CALCULATING AND DISPLAYING EARTHQUAKE FAULT-PLANE SOLUTIONS,
by P. REASENBERG and D. OPPENHEIMER, U.S. GEOLOGICAL SURVEY OPEN-FILE REPORT 85-77?)

REQUIRES ROUTINES: CALCOMP STYLE PLOT ROUTINES PLOTS, PLOT, NEWPEN, SYMBOL.

DEPARTURES FROM FORTAN-77 STANDARD:
1. KEYWORD "READONLY" IN OPEN STATEMENT
2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS
3. ARGUMENT LIST BUILT-IN FUNCTION "XREF"
4. CALLS TO VAX SYSTEM ROUTINES "SYSSBINTM", "SYSSCHDWK", "SYSSHIBER" (SUBROUTINE DELAY)

OUTPUT: GRAPHIC OUTPUT ONLY

AUTHORS: PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK. SOME OF THE ROUTINES
WERE ADOPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.

REAL AIN ! RAY ANGLE OF INCIDENCE
REAL AZM ! RAY AZIMUTH
REAL CX ! X POSITION OF STEREO NET CENTER
REAL CMAX ! GREATEST X POSITION IN PLOT
REAL CY ! Y POSITION OF STEREO NET CENTER
REAL DAI ! DIP ANGLE OF PRINCIPLE PLANE
REAL DA2 ! DIP ANGLE OF AUXILIARY PLANE
REAL DD1 ! DIP DIRECTION OF PRINCIPLE PLANE
REAL DD2 ! DIP DIRECTION OF AUXILIARY PLANE
CHARACTER*1 DISC ! FLAG: IF "*" THEN FIRST MOTION DISCREPANT WITH SOLUTION
REAL DIST ! EPICENTRAL DISTANCE
REAL DX ! INCREMENTAL X POSITION BETWEEN BEACHBALLS
REAL DY ! INCREMENTAL Y POSITION BETWEEN BEACHBALLS
CHARACTER*132 EVENT ! HYP0P. SUMMARY- CARD
CHARACTER*5 FILLAM ! FILE NAME OF DATA
CHARACTER*1 HEAD ! HEADER FLAG: N= PLOT EVENT NUMBER, D= EVENT DATE
REAL HITE ! HEIGHT OF EVENT #
INTEGER I ! LOOP INDEX OVER NUMBER OF LINES OF NEARBY SOLUTIONS
INTEGER ILINE ! INPUT LINE NUMBER
INTEGER IOS ! IO STATUS DESCRIPTOR
CHARACTER*40 LINE ! SCRATCH VARIABLE FOR PLOT OUTPUT
CHARACTER*1 MULT ! FLAG: Y(N)=DO (NOT) PLOT MULTIPLE SOLUTIONS
INTEGER MAXEVNT ! MAXIMUM NUMBER OF EVENTS PER PAGE
CHARACTER*4 NAME ! STATION NAME
INTEGER NCHAR ! NUMBER OF CHARACTERS TO BE PLOTTED
INTEGER NEV ! EVENT NUMBER
INTEGER NLABEL ! CURRENT EVENT LABEL NUMBER
INTEGER NLINE ! NUMBER OF LINES OF NEARBY SOLUTIONS
INTEGER NSKIP ! NUMBER OF EVENTS TO SKIP
INTEGER NSTAR ! NUMBER OF SOLUTIONS WITH FIT WITHIN 5% OF BEST SOLUTION
INTEGER NUM ! NUMBER OF EVENTS PLOTTED
INTEGER NY ! NUMBER OF BEACHBALLS IN Y DIRECTION
REAL PI ! PI
REAL PLOTLN ! PLOT LENGTH IN X DIRECTION
CHARACTER*1 PLTPOL ! FLAG: Y(N)=DO (NOT) PLOT FIRST MOTION DATA
CHARACTER*4 PRMK ! FIRST MOTION DESCRIPTION (e.g. IPU0)
REAL RAD ! RADIUS OF STEREO NET
REAL RHAX1 ! RADIUS OF FIRST MOTION SYMBOL
REAL RHAX2 ! RAKE OF AUXILIARY PLANE
REAL SAI
REAL SA2  ! RAKE OF AUXILLIARY PLANE
CHARACTER*1 STAR  ! MULTIPLE INDICATOR
CHARACTER*1 SYM  ! FIRST MOTION DIRECTION
CHARACTER*80 TITLE  ! DATA SET DESCRIPTOR
REAL WT  ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
REAL XPOS  ! X PLOT POSITION
REAL YPOS  ! Y PLOT POSITION

C
PARAMETER (HUE = 0.15, MXEVNT = 42, RMAX1 = 0.5, RMAX2 = .065,
& YPAGE = 18.5)
PI = ATAN(1.0)*4.0 .
RAD = PI/180.0
DY = RMAX1*3
NY = YPAGE/DY
DX = DY
C
1# WRITE (6, 15) 'ENTER NAME OF DATA FILE: '
15 FORMAT ('$', A)
READ 15, 16, ERR = 18) FILNAM
16 FORMAT (A)
OPEN UNIT = 2, FILE = FILNAM, ERR = 19, STATUS = 'OLD', BLANK =
& 'ZERO', READONLY)
WRITE (6, 15) 'DO YOU WANT TO PLOT EVENT HEADERS AS NUMBERS (N) OR
& DATES (D)? '
2# READ 15, 16, IOSTAT = IOS) HEAD
IF (HEAD .EQ. 'Y') HEAD = 'Y'
IF (HEAD .EQ. 'N') HEAD = 'N'
IF ((HEAD .NE. 'N') .AND. HEAD .NE. 'D') .OR. IOS .NE. 0) THEN
PRINT *, '***** PLEASE ANSWER "N" OR "D"; TRY AGAIN ******
GOTO 2#
END IF
WRITE (6, 15) 'ENTER NUMBER OF MECHANISMS TO SKIP (INCLUDING MULTI
PLE SOLUTIONS)!'
3# READ 15, *, IOSTAT = IOS) NSKIP
IF (NSKIP .LT. 0 .OR. IOS .NE. 0) THEN
PRINT *, '***** INVALID NUMBER; TRY AGAIN ******
GOTO 3#
END IF
WRITE (6, 15) 'DO YOU WANT TO PLOT MULTIPLE SOLUTIONS (Y OR N)? '
4# READ 15, *, IOSTAT = IOS) MULT
IF (MULT .EQ. 'Y') MULT = 'Y'
IF (MULT .EQ. 'N') MULT = 'N'
IF ((MULT .NE. 'Y') .AND. MULT .NE. 'N') .OR. IOS .NE. 0) THEN
PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN ******
GOTO 4#
END IF
WRITE (6, 15) 'DO YOU WANT TO PLOT FIRST MOTION DATA (Y OR N)? '
5# READ 15, 16, IOSTAT = IOS) PLTPOL
IF (PLTPOL .EQ. 'Y') PLTPOL = 'Y'
IF (PLTPOL .EQ. 'N') PLTPOL = 'N'
IF ((PLTPOL .NE. 'Y') .AND. PLTPOL .NE. 'N') .OR. IOS .NE. 0) THEN
PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN ******
GOTO 5#
END IF
C
C READ HYP071 HEADER CARD (FIRST LINE IN MODEL FILE)
C
MLABEL = 0
NEV = 0
NUM = 0
CXXMAX = 0
ILINE = 1
READ (2, 68, ERR = 2000) TITLE
6# FORMAT (A)
C INITIALIZE PLOT PROGRAM
C
CALL PLOTS (0., 0., 0.)
CALL ERASE
CALL DELAY ('0000 00:00:01.00')
CALL PLOT (.1., .1., -3)
CALL NEWPEN (2)

C READ EVENT
C
7B  ILINE = ILINE + 1
READ (2, 80, END = 1000, ERR = 2000) EVENT
80  FORMAT (IX, A132)
NEV = NEV * 1
READ (EVENT, 90, ERR = 2000) DDL DA1, SA1, STAR
90  FORMAT (T81, F4.0, F3.0, F4.0, T132, A1)
IF (STAR .EQ. ' ') NLABEL = NLABEL + 1
IF (NEV .GT. NSK1P) .AND. (MULT .EQ. 'Y') THEN
END PLOT IF MORE THAN MXEVNT EVENTS
NUM = NUM + 1
IF (NUM .GT. MXEVNT) THEN
NUM = 1
CALL PLOT (0., 0., -999)
CALL ERASE
CALL DELAY ('0000 00:00:01.00')
CALL PLOT (.1., .1., -3)
CALL NEWPEN (2)
END IF
CY = YPAGE - DY*FLOAT(JMOD(NUM - 1, NY) - RMAX1*2.0
CX = RMAX1 + DX*FLOAT((NUM - 1)/NY)
IF (JMOD(NUM - 1, NY) .EQ. 0) CXMAX = CXMAX + DX
XPOS = CX - RMAX1
YPOS = CY * RMAX1 * .I
IF (HEAD .EQ. 'N') THEN
IF (NLABEL .LT. 10) THEN
NCHAR = 2
WRITE (LINE, '(II, A1)') NLABEL, STAR
ELSE IF (NLABEL .GE. 10 .AND. NLABEL .LT. 100) THEN
NCHAR = 3
WRITE (LINE, '(II2, A1)') NLABEL, STAR
ELSE
NCHAR = 4
WRITE (LINE, '(II3, A1)') NLABEL, STAR
END IF
CALL SYMBOL (XPOS, YPOS, HITE, XREF(LINE), 0., NCHAR)
ELSE
WRITE (LINE, '(A11, A1)') EVENT(1:11), STAR
NCHAR = 12
CALL SYMBOL (XPOS, YPOS, HITE*.75, XREF(LINE), 0., NCHAR)
END IF
C PLOT STEREO NET PERIMETERS
C
CALL STRNT1 (CX, CY, RAD, RMAX1)
C
CALL PLOTPL (CX, CY, DA1, PI, RAD, RMAX1, DD1 - 90.)
CALL AUXPLN (DDL DA1, SA1, DD2, DA2, SA2)
CALL PLOTPL (CX, CY, DA2, PI, RAD, RMAX1, DD2 - 90.)
CALL TPPLOT (CX, CY, DA1, DDI, RMAX2*1.5, PI, PLTPOL, RAD, RMAX1, SAI, 1.0)
END IF

C SKIP OVER NEARBY SOLUTIONS
C
ILINE = ILINE + 1
READ (2, *, ERR = 2000) NSTAR
NLINE = (NSTAR-1)/11 + 1
DO 110 I = I, NLINE
ILINE = ILINE + 1
READ (2, 100, ERR = 2000) LINE
100 FORMAT (A)
110 CONTINUE
C
C READ PHASE CARD
C
120 ILINE = ILINE + 1
READ (2, 130, END = 1000, ERR = 2000) NAME, DIST, AZM, A14, PRMK, WT, DISC
130 FORMAT (IX, A4, 3F6.1, 3X, A4, F5.2, 2X, A1)
IF (NAME .NE. ' ') THEN
  IF (MULT .EQ. 'N' .AND. STAR .EQ. ' ') THEN
    SYM = 'C'
  ELSE
    SYM = 'D'
  END IF
  CALL PLTSYM (A14, AZM, CX, CY, RMAX2, ' ', PI, RAD, RMAX1, SYM, 1.0)
END IF
GOTO 120
GOTO 70
C
C END OF FILE
C
1000 CALL PLOT (0., 0., 999)
PLOTLN = CXMAX + RMAX1
PRINT *, 'MAXIMUM PLOT SIZE =', PLOTLN, 'X', YPAGE
CLOSE (2)
STOP
C
C READ ERROR
C
2000 PRINT *, 'READ ERROR ON LINE', ILINE
STOP
END

C
C
C
C SUBROUTINE AUXPLN (DD1, DAI, SAI, DDI2, DA2, SAI2)
C
C CALCULATE THE AUXILIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.
C
C WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)
C
C ANGLE VARIABLES PHI, DEL AND LAM ARE AS DEFINED IN AKI AND RICHARDS. (1980), P.114.
REAL DA1, DD1, SA1, DA2, DD2, SA2

! DIP ANGLE IN DEGREES
I DIP DIRECTIONS IN DEGREES
! SLIP ANGLE IN DEGREES
I DIP ANGLE OF AUXILIARY PLANE
! DIP DIRECTION OF AUXILIARY PLANE
! SLIP ANGLE OF AUXILIARY PLANE

DOUBLE PRECISION BOT
DOUBLE PRECISION DEL1
LOGICAL FIRST
DOUBLE PRECISION PHI1
DOUBLE PRECISION PHIZ
DOUBLE PRECISION RAD
DOUBLE PRECISION SGN
DOUBLE PRECISION TOP
DOUBLE PRECISION XLAM1
DOUBLE PRECISION XLAM2

! SCRATCH VARIABLE
I DIP ANGLE OF PRINCIPAL PLANE IN RADIANS
! TEST: TRUE IF FIRST TIME INTO ROUTINE
! FAULT PLANE STRIKE OF PRINCIPAL PLANE
! STRIKE OF AUXILIARY PLANE IN RADIANS
! CONVERSION FACTOR FROM DEGREES TO RADIAN
! SAVES PRINCIPAL PLANE SLIP ANGLE FOR ASSIGNING PROPER SIGN TO AUXILIARY
! SCRATCH VARIABLE
! SLIP ANGLE OF PRINCIPAL PLANE IN RADIANS
I SLIP ANGLE OF AUXILIARY PLANE

DATA FIRST /.TRUE./
SAVE FIRST, RAD

IF (FIRST) THEN
  FIRST = .FALSE.
  RAD = DATAN(1.000)/45.000
END IF

PHI1 = DD1 - 90.000
IF (PHI1 .LT. 0.000) PHI1 = PHI1 + 360.000
PHI1 = PHI1/RAD
DELI = DA1*RAD
SGN = SAI
XLAM1 = SAI*RAD

TOP = DCOS(XLAM1)*DSIN(PHI1) - DCOS(DELI)*DSIN(XLAM1)*DCOS(PHI1)
BOT = DCOS(XLAM1)*DCOS(PHI1) + DCOS(DELI)*DSIN(XLAM1)*DSIN(PHI1)
DD2 = DATAN2(TOP, BOT)/RAD
PHIZ = (DD2 - 90.000)*RAD
IF (SAI .LT. 0.000) DD2 = DD2 - 180.000
IF (DD2 .LT. 0.000) DD2 = DD2 + 360.000
IF (DD2 .GT. 360.000) DD2 = DD2 - 360.000

DA2 = DACOS(DSIN(DABS(XLAM2))*DSIN(DELI)/RAD
XLAM2 = -DCOS(PHI2)*DSIN(DELI)*DSIN(PHI1) + DSIN(PHI2)*DSIN(DELI)*DCOS(PHI1)

C MACHINE ACCURACY PROBLEM

IF (XLAM2 .GT. 1.000) THEN
  XLAM2 = 1.000
END IF

RETURN
END

C SUBROUTINE CIRCLE (SIZE, TWOPI, X#, Y#)
C PLOT A CIRCLE
C
REAL SIZE
REAL TWOPI
REAL X0
REAL Y0

C REAL ANGLE
INTEGER J
INTEGER N
REAL SIZE2
REAL X
REAL Y

SIZE2 = SIZE*0.5
C COMPUTE OPTIMUM # OF POINTS TO DRAW
C
N = 2*SORT(SIZE2*2.)
IF (N .LT. 10) N = 10
C DRAW CIRCLE
C
X = X0 + SIZE2
CALL PLOT (X, Y0, 3)
DO 10 J = 1, N
  ANGLE = TWOPI*FLOAT(J)/FLOAT(N)
  X = X0 + SIZE2*COS(ANGLE)
  Y = Y0 + SIZE2*SIN(ANGLE)
  CALL PLOT (X, Y, 2)
10 CONTINUE
C
RETURN
END

SUBROUTINE DELAY (TIME)
C CREATES A DELAY FOR SCREEN RECOVERY
C
CHARACTER*(*)(TIME
DOUBLE PRECISION B
C CONVERT ASCII TIME TO BINARY TIME
CALL SYS$BINTIM (TIME,B)
C SCHEDULE A WAKEUP FOR A ELAPSED TIME (NEGATIVE B)
CALL SYS$SCHDWK (,-B.)
C Hibernate AND ReAwake
CALL SYS$HIBER
C
RETURN
END
SUBROUTINE ERASE
C SENDS A SCREEN ERASE CODE TO A TEKTRONIX TERMINAL
C
CHARACTER*1 A
CHARACTER*1 B
C
A = CHAR(27)
B = CHAR(12)
WRITE (6, 10) A, B
10 FORMAT (IX, 2AI)
RETURN
END

SUBROUTINE PLOTPL (CX, CY, DPIDG, PI, RAD, RMAX, STRKDG)
C PLOTS FAULT PLANE ON LOWER HEMISPHERE STEREO NET
C
REAL CX
REAL CY
REAL DPIDG
REAL PI
REAL RAD
REAL RMAX
REAL STRKDG
REAL ANG
REAL AZPI91
REAL ARG
REAL AZ
REAL CON
REAL DIPRD
INTEGER I
INTEGER MI
REAL RADIUS
REAL SAZ(91)
REAL STRKRD
REAL TAZ
REAL TPD
REAL X
REAL Y
C
STRKRD = STRKDG*RAD
DIPRD = DPIDG*RAD
TPD = TAN(PI*.5 - DIPRD)**2
C
CASE OF VERTICAL PLANE
C
IF (DPIDG .EQ. 90.0) THEN
X = RMAX*SIN(STRKRD) + CX
Y = RMAX*COS(STRKRD) + CY
CALL PLOT (X, Y, 3)
X = RMAX*SIN(STRKRD + PI) + CX
Y = RMAX*COS(STRKRD + PI) + CY
CALL PLOT (X, Y, 2)
RETURN
END IF
C
C COMPUTE ANGLE OF INCIDENCE, AZIMUTH
C
DO 10 I = 1, 90
  ANG = FLOAT(I) - I*RAD
  ARG = SQRT((COS(DIPRD)**2)*(SIN(ANG)**2))/COS(ANG)
  SAZ(I) = ATAN(ARG)
  TAZ = TAN(SAZ(I))**2
  ARG = SQRT(TPD + TPD*TAZ + TAZ)
  AIMP(I) = ACOS(TAN(SAZ(I))/ARG)
10 CONTINUE
SAZ(91) = 90.*RAD
AIMP(91) = PI*.5 - DIPRD
C
C PLOT PLANE
C
CON = RMAX*SORT(2.)
DO 20 I = 1, 180
  IF (I .LE. 91) THEN
    MI = I
    AZ = SAZ(I) + STRKRD
  ELSE
    MI = 181 - I
    AZ = PI - SAZ(MI) + STRKRD
  END IF
  RADIUS = CON*SIN(AIMP(MI)*#5)
  X = RADIUS*SIN(AZ) + CX
  Y = RADIUS*COS(AZ) + CY
  IF (I .EQ. 1) THEN
    CALL PLOT (X, Y, 3)
  ELSE
    CALL PLOT (X, Y, 2)
  END IF
20 CONTINUE
C
RETURN
C
C
C SUBROUTINE STRN1 (CX, CY, RAD, RMAX)
C
C PLOT PERIMETER OF A STEREO NET
C
REAL       CX       I X POSITION OF CIRCLE CENTER
REAL       CY       I Y POSITION OF CIRCLE CENTER
REAL       RAD      I PI/100
REAL       RMAX     I RADIUS OF CIRCLE
REAL       CSIZ     I SCRATCH VARIABLE (RMAX/100)
INTEGER    I        I LOOP INDEX OVER DEGREES
INTEGER    N        I TESTS 10 DEGREE TICK POSITION
INTEGER    NN       I TESTS 90 DEGREE TICK POSITION
REAL       P        I TANGENT LENGTH
REAL       PHI      I AZIMUTH IN RADIANS
REAL       X        I X POSITION OF CIRCLE
REAL       XP       I X POSITION OF END OF TICK
REAL       Y        I Y POSITION OF CIRCLE
REAL       YP       I Y POSITION OF END OF TICK
C
CALL NEWPEN (2)
C
C DRAW CIRCLE @ 5 DEGREE INCREMENTS
DO 10 I = 1, 73
PHI = FLOAT(I - 1)*RAD*5.0
X = RMAX*COS(PHI) + CX
Y = RMAX*SIN(PHI) + CY
N = (I - 1) - ((I - 1)/10)*10
NN = (I - 1) - ((I - 1)/50)*50
IF ((N .EQ. 0) .AND. (I .GT. 10)) THEN
  P = 0.02*RMAX
ELSE IF ((NN .EQ. 0) .AND. (I .GT. 90)) THEN
  P = 0.04*RMAX
ELSE
  P = 0.01*RMAX
END IF
XP = (RMAX + P)*COS(PHI)*CX
YP = (RMAX + P)*SIN(PHI)*CY
IF (I .GT. 1) THEN
  CALL PLOT (X, Y, 2)
  IF (JMOD(I - 1, IB) .EQ. 0) THEN
    XP = (RMAX + TIC)*COS(PHI)*CX
    YP = (RMAX + TIC)*SIN(PHI)*CY
    CALL PLOT (XP, YP, 2)
    CALL PLOT (X, Y, 3)
  END IF
ELSE
  CALL PLOT (X, Y, 3)
END IF
CONTINUE

C PLOT AT CENTER
CSIZ = .01*RMAX
CALL PLOT (CX - CSIZ, CY, 3)
CALL PLOT (CX + CSIZ, CY, 3)
CALL PLOT (CX, CY - CSIZ, 3)
CALL PLOT (CX, CY + CSIZ, 2)
RETURN

END

SUBROUTINE TPLOT (CX, CY, DAI, DD1, HITE, PI, PLTPOL, RAD, RMAX, & SAI, WT)
PLOT P AND T AXES

REAL CX                      I X POSITION OF CIRCLE CENTER
REAL CY                      I Y POSITION OF CIRCLE CENTER
REAL DAI                     I DIP ANGLE
REAL DD1                     I DIP DIRECTION
REAL HITE                    I HEIGHT OF P.T SYMBOL
REAL PI                      I PI
CHARACTER*1 PLTPOL           I FLAG: Y(N)=DO (NOT) PLOT FIRST MOTION DATA
REAL RAD                     I PI/180
REAL RMAX                    I RAD/US OF CIRCLE
REAL SAI                     I RAKE
REAL WT                      I WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
REAL AINI                    I ANGLE OF INCIDENCE OF P/T AXIS
REAL AIN2                    I ANGLE OF INCIDENCE OF T/P AXIS
REAL ANG                     I ANGLE OF PLOT SYMBOL
REAL AZ1        ! AZIMUTH OF P/T AXIS
REAL AZ2        ! AZIMUTH OF T/P AXIS
CHARACTER*4 BLANK        ! BLANK
REAL DA2        ! DIP ANGLE OF AUXILIARY PLANE
REAL DD2        ! DIP DIRECTION OF AUXILIARY PLANE
REAL SA2        ! STRIKE OF AUXILIARY PLANE
CHARACTER*1 SYMI        ! P/T PLOT SYMBOL
CHARACTER*1 SYM2        ! T/P PLOT SYMBOL

PARAMETER (ANG = 0.0, BLANK = ' ')

FIND AUXILIARY PLANE
CALL AUXPLN (DD1, DAI, SA1, DD2, DA2, SA2)

FIND P AND T AXES
CALL TANP (AIN1, AIN2, AZ1, AZ2, DAI, DA2, DD1, DD2, PI, RAD)
IF (SA1 .LT. 0.) THEN
  SYMI = 'P'
  SYM2 = 'T'
ELSE
  SYMI = 'T'
  SYM2 = 'P'
END IF

PLOT SYMBOLS
IF (SYMI .EQ. 'T' .OR. PLTPOL .EQ. 'Y') CALL PLTSYM (AIN1, AZ1, & CX, CY, HITE, BLANK, PI, RAD, RMAX, SYMI, WT)
IF (SYM2 .EQ. 'T' .OR. PLTPOL .EQ. 'Y') CALL PLTSYM (AIN2, AZ2, & CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM2, WT)

RETURN
END

SUBROUTINE TANP (AIN1, AIN2, AZ1, AZ2, DAI, DA2, DD1, DD2, PI, RAD)
GIVEN TWO PLANES COMPUTE AZ AND ANGLE OF INCIDENCE OF P & T AXES

REAL AIN1        ! ANGLE OF INCIDENCE OF P/T AXIS
REAL AIN2        ! ANGLE OF INCIDENCE OF T/P AXIS
REAL AZ1        ! AZIMUTH OF P/T AXIS
REAL AZ2        ! AZIMUTH OF T/P AXIS
REAL DAI        ! DIP ANGLE OF PRINCIPLE PLANE
REAL DA2        ! DIP ANGLE OF AUXILIARY PLANE
REAL DD1        ! DIP DIRECTION OF PRINCIPLE PLANE
REAL DD2        ! DIP DIRECTION OF AUXILIARY PLANE
REAL PI        ! PI
REAL RAD        ! PI/180

REAL ALAT1        ! DIP ANGLE IN RADIANS OF PRINCIPLE PLANE MEASURED FROM VERTICAL
REAL ALAT2        ! DIP ANGLE IN RADIANS OF AUXILIARY PLANE MEASURED FROM VERTICAL
REAL ALON1        ! DDI IN RADIANS
REAL ALON2        ! DD2 IN RADIANS
REAL AZIMTH        ! AZIMUTH IN RADIANS OF POLE ??
REAL AZB        ! AZIMUTH FROM POLE OF AUXILIARY PLANE TO POLE OF PRINCIPLE ??
REAL BAZM        ! NOT USED
REAL DELTA        ! NOT USED
REAL PLUNGE        ! PLUNGE IN RADIANS OF POLE ??
REAL SHIF\T  ! AZIMUTHAL SHIFT FROM POLE OF PLANE TO P TO T AXIS (= 45 DEGREES)?
REAL XPOS   ! NOT USED
REAL YPOS   ! NOT USED

C
PARAMETER (SHIFT = .7853981)
C
ALAT1 = (90. - DAI)*RAD
ALON1 = DDI*RAD
ALAT2 = (90. - DAI)*RAD
ALON2 = DDI*RAD
CALL REFT (ALAT2, ALON2)
CALL DELAZ (ALAT1, ALON1, DELTA, AZ#, BAZM, XPOS, YPOS)
CALL BACK (SHIFT, AZ#, PLUNGE, AZIMTH)
IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.*PI, AZIMTH)
AZI = AZIMTH/RAD
A1N1 = PLUNGE/RAD + 90.
AZ# = AZ# + PI
CALL BACK (SHIFT, AZ#, PLUNGE, AZIMTH)
IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.*PI, AZIMTH)
AZ# = AZIMTH/RAD
A1N2 = PLUNGE/RAD + 90.

C
RETURN
END

C

SUBROUTINE PLTSYM (AIN, AZ, CX, CY, HITE, NAME, PI, RAD, RMAX, & SYM, WT)
C
C PLOT EITHER FIRST MOTION SYMBOL (C,D,+,--) WITH STATION NAME NEXT TO SYMBOL, OR STRESS AXES SYMBOL (P & T)
C
C
REAL AIN            ! ANGLE OF INCIDENCE OF SYMBOL
REAL AZ             ! AZIMUTH OF SYMBOL
REAL CX             ! X POSITION OF CIRCLE CENTER
REAL CY             ! Y POSITION OF CIRCLE CENTER
REAL HITE           ! HEIGHT OF SYMBOL
REAL NAME           ! STRING TO BE PLOTTED TO RIGHT OF SYMBOL
REAL PI             ! PI
REAL RAD            ! 1/180
REAL RMAX           ! RADIUS OF CIRCLE
REAL SYM            ! PLOT SYMBOL
REAL WT             ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
REAL AINR           ! AIN IN RADIANS
REAL ANG            ! ANGLE OF CIRCLE BISECTOR FOR COMPRESSINAL FILL
REAL ANGLE          ! SAME AS ANG
REAL AZR            ! AZ IN RADIANS
REAL CON            ! RMAX = SORT(2.#)
REAL DANG           ! ANGLE INCREMENT OF CIRCLE BISECTOR FOR COMPRESSINAL FILL
LOGICAL FIRST       ! FLAG FIRST TIME INTO ROUTINE
LOGICAL R           ! DISTANCE FROM CX, CY TO PLOT POSITION
REAL X              ! X POSITION OF SYMBOL
REAL XPOS           ! X POSITION OF CIRCLE BISECTOR FOR COMPRESSINAL FILL
REAL Y              ! Y POSITION OF SYMBOL
REAL YPOS           ! Y POSITION OF CIRCLE BISECTOR FOR COMPRESSINAL FILL
C
AZ# = AZ*RAD
A1N# = A1N*RAD
C
C UPGOING RAYS
C
IF (A1N .GT. 90.) THEN  
   AINR = PI - AINR  
   AZR = PI + AZR  
END IF  
CON = RMAX*SORT(2.0)  
R = CON*SIN(AINR*.5)  
X = R*SIN(AZR) + CX  
Y = R*COS(AZR) + CY  
C STRESS AXIS SYMBOL  
C  
   IF ((SYM .EQ. 'P') .OR. (SYM .EQ. 'T')) THEN  
      X = X - .286*HITE  
      Y = Y + .5*HITE  
      CALL SYMBOL (X, Y, HITE, XREF(SYM), #.#, 1)  
   ELSE  
      C  
      C FIRST MOTION SYMBOL  
      C  
      CALL CIRCLE (HITE, 2.0*PI, X, Y)  
      C  
      C FILL IN COMPRESSION CIRCLE  
      C  
      IF (SYM .EQ. 'C') THEN  
         N = 9  
         DANG = PI/FLOAT(N)  
         DO 10 I = 1, N  
            ANG = ANG + DANG  
            XPOS = X + .5*HITE*COS(ANGLE)  
            YPOS = Y + .5*HITE*SIN(ANGLE)  
            CALL PLOT (XPOS, YPOS, 3)  
            ANG = ANG + PI  
            XPOS = X + .5*HITE*COS(ANGLE)  
            YPOS = Y + .5*HITE*SIN(ANGLE)  
            CALL PLOT (XPOS, YPOS, 2)  
   10 CONTINUE  
   END IF  
C  
C PLOT STATION NAME  
C  
   IF (NAME .NE. ' ') CALL SYMBOL (X + HITE/2., Y, HITE/2.,  
   & XREF(NAME), #, 4)  
END IF  
RETURN  
END  
C  
C SUBROUTINE GEOCEN  
C  
GEOCEN - CALCULATE GECENTRIC POSITIONS, DISTANCES, AND AZIMUTHS (BRUCE JULIAN, USGS MENLO PARK, CA)  
C  
THE GECENTRIC DISTANCE DELTA AND AZIMUTH AZ0 FROM POINT (LAT0, LON0) TO POINT (LAT1, LON1) ARE CALCULATED FROM  
COS(DELTA) = COS(LAT0')*COS(LAT1') + COS(LON1 - LON0) + SIN(LAT0')*SIN(LAT1')  
SIN(DELTA) = SQRT(A*A + B*B)  
TAN(AZ0) = A/B  
C  
WHERE  
A = COS(LAT0')*SIN(LON1-LON0)
B = \cos(LAT_0') \times \sin(LAT_1') - \sin(LAT_0') \times \cos(LAT_1') \times \cos(LON_1 - LON_0)

LAT_0', LAT_1' = GEOCENTRIC LATITUDES OF POINTS
LON_0, LON_1 = LONGITUDES OF POINTS

THE GEOCENTRIC LATITUDE LAT' IS OBTAINED FROM THE GEOGRAPHIC LATITUDE LAT BY
TAN(LAT') = (1 - Alpha) * (1 - Alpha) * TAN(LAT)
WHERE Alpha IS THE FLATTENING OF THE ELLIPSOID. SEE FUNCTION GGTGC FOR CONVERSION.

THE BACK AZIMUTH IS CALCULATED BY THE SAME FORMULAS WITH (LAT_0', LON_0) AND
(LAT_1', LON_1) INTERCHANGED.
AZIMUTH IS MEASURED CLOCKWISE FROM NORTH THRU EAST.

REAL R, Theta
REAL AZ0, AZ1
REAL CDelt
REAL COLAT
REAL CT0, CT1
REAL CZ0
REAL Delta
REAL Dlon
REAL ERad
REAL FLAT

INTEGER LAMBDA

RADIUS, AZIMUTH IN POLAR COORDINATES
AZIMUTH FROM REFERENCE POINT TO SECONDARY POINT IN RADIANS
AZIMUTH FROM SECONDARY POINT TO REFERENCE POINT IN RADIANS
SINE OF DELTA TO SECONDARY POINT
COSINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
SINE OF REFERENCE POINT LATITUDE
SINE OF SECONDARY POINT LATITUDE
COSINE OF AZIMUTH TO SECONDARY POINT
GEOCENTRIC DISTANCE IN DEGREES
AZIMUTH IN POLAR COORDINATES
EQUATORIAL RADIUS (CHOVITZ, 1981, EOS, 62, 65-67)

PARAMETER (FLAT = 1./298.257, ERAD = 6378.137)
PARAMETER (LAMBDA = FLAT^2 - FLAT) / (1. - FLAT)**2

STEP: STORE THE GEOCENTRIC COORDINATES OF THE REFERENCE POINT
ENTRY STEPLAT, LON)
ST0 = COS(LAT)
CT0 = SIN(LAT)
PHI0 = LON
RETURN

DELAZ - CALCULATE THE GEOCENTRIC DISTANCE, AZIMUTHS
ENTRY DELAZ(LAT, LON, DELTA, AZ0, AZ1, X, Y)
CT1 = SIN(LAT)
ST1 = COS(LAT)

IF (CT1 - CT0)  .EQ. 0. .AND. (LON - PHI0)  .EQ. 0. ) THEN
DELTA = 0.
AZ0 = 0.
AZ1 = 0.
ELSE
DOLON = SIN(LON - PHI0)
COLON = COS(LON - PHI0)
CDelt = ST1*ST1*COLON + CT0*CT1
CALL CVRTOP (ST0, CT0, SDOLN, STI, SDLON, SDELT, AZ0)

DELTA = ATAN2 (SDELT, CDELT)
CALL CVRTOP (STI, CTI, SDOLN, SDOLN, ST0, SDLON, SDELT, AZ1)
IF (AZ0 .LT. 0.0) AZ0 = AZ0 + TWOPI
IF (AZI .LT. 0.0) AZI = AZI + TWOPI
END IF

COLAT = PI/2. - (LAT - OLAT)/2.
RADIUS = ERAD/SORTC (I, #) + LAMBDACOS(COLAT)**2
V = RADIUS*DELTA*COS(AZ0)
X = RADIUS*DELTA*SIN(AZ0)
RETURN

C BACK - CALCULATE GEOCENTRIC COORDINATES OF SECONDARY POINT FROM DELTA, AZ
C
ENTRY BACK (DELTA, AZ0, LAT, LON)
C
SDELT = SIN(DELTA)
CDELT = COS(DELTA)
CZ0 = COS(AZ0)
CTI = ST0*SDELT*CZ0 + CT0*CDELT
CALL CVRTOP (ST0, CDELT - CT0*SDELT - CZ0, SDOLN, STI, SDLON)
LAT = ATAN2 (CTI, STI)
LON = PHI0 + DLON
IF (ABS(LON)) .GT. PI) LON = LON - SIGN(TWOPI, LON)
RETURN

C

C SUBROUTINE CVRTOP (X, Y, R, THETA)
C
CVRTOP - CONVERT FROM RECTANGULAR TO POLAR COORDINATES (BRUCE JULIAN, USGS MENLO PARK, CA)
C
REAL X, Y, R, THETA
I X, Y RECTANGULAR COORDINATES
I RADIUS, AZIMUTH IN POLAR COORDINATES
R = SQRT (X**X + Y**Y)
THETA = ATAN2 (Y, X)
RETURN
END